Min-Max Graph Partitioning and Small Set Expansion

Nikhil Bansal[*]   Uriel Feige[†]   Robert Krauthgamer[‡]   Konstantin Makarychev[§]

Viswanath Nagarajan[¶]   Joseph (Seffi) Naor[‖]   Roy Schwartz[**]

October 21, 2011

Abstract

We study graph partitioning problems from a *min-max* perspective, in which an input graph on n vertices should be partitioned into k parts, and the objective is to minimize the maximum number of edges leaving a single part. The two main versions we consider are where the k parts need to be of equal-size, and where they must separate a set of k given terminals. We consider a common generalization of these two problems, and design for it an $O(\sqrt{\log n \log k})$-approximation algorithm. This improves over an $O(\log^2 n)$ approximation for the second version due to Svitkina and Tardos [ST04], and roughly $O(k \log n)$ approximation for the first version that follows from other previous work. We also give an improved $O(1)$-approximation algorithm for graphs that exclude any fixed minor.

Our algorithm uses a new procedure for solving the Small-Set Expansion problem. In this problem, we are given a graph G and the goal is to find a non-empty set $S \subseteq V$ of size $|S| \leq \rho n$ with minimum edge-expansion. We give an $O(\sqrt{\log n \log (1/\rho)})$ bicriteria approximation algorithm for the general case of Small-Set Expansion, and $O(1)$ approximation algorithm for graphs that exclude any fixed minor.

1 Introduction

We study graph partitioning problems from a *min-max* perspective. Typically, graph partitioning problems ask for a partitioning of the vertex set of an undirected graph under some problem-specific constraints on the different parts, e.g., balanced partitioning or separating terminals, and the objective is *min-sum*, i.e., minimizing the total weight of the edges connecting different parts. In the *min-max* variant of these problems, the goal is different — minimize the weight of the edges leaving a single part, taking the maximum over the different parts. A canonical example, that we consider throughout the paper, is the Min–Max k–Partitioning problem: given an undirected graph $G = (V, E)$ with nonnegative edge-weights and $k \geq 2$, partition the vertices into k (roughly) equal

---

[*]Eindhoven University of Technology, The Netherlands. E-mail: `n.bansal@tue.nl`

[†]Weizmann Institute of Science, Rehovot, Israel. Work supported in part by The Israel Science Foundation (grant #873/08). Email: `uriel.feige@weizmann.ac.il`

[‡]Weizmann Institute of Science, Rehovot, Israel. Work supported in part by The Israel Science Foundation (grant #452/08), and by a Minerva grant. Email: `robert.krauthgamer@weizmann.ac.il`

[§]IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. Email: `konstantin@us.ibm.com`

[¶]IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. Email: `viswanath@us.ibm.com`

[‖]Computer Science Dept., Technion, Haifa, Israel. Email: `naor@cs.technion.ac.il`

[**]Computer Science Dept., Technion, Haifa, Israel. Email: `schwartz@cs.technion.ac.il`

1

parts $S_1, \ldots, S_k$ so as to minimize $\max_i \delta(S_i)$, where $\delta(S)$ denotes the sum of edge-weights in the cut $(S, V \setminus S)$. We design a bicriteria approximation algorithm for this problem. Throughout, let $w : E \to \mathbb{R}^+$ denote the edge-weights and let $n = |V|$.

Min-max partitions arise naturally in many settings. Consider the following application in the context of cloud computing, which is a special case of the general graph-mapping problem considered in [BLNZ11] (and also implicit in other previous works [YYRC08, ZA06, CRB09]). There are n processes communicating with each other, and there are k machines, each having a bandwidth capacity C. The goal is to allocate the processes to machines in a way that balances the load (roughly n/k processes per machine), and meets the outgoing bandwidth requirement. Viewing the processes as vertices and the traffic between them as edge-weights, we get the Min–Max k–Partitioning problem. In general, balanced partitioning (either min-sum or min-max) is at the heart of many heuristics that are used in a wide range of applications, including VLSI layout, circuit testing and simulation, parallel scientific processing, and sparse linear systems.

Balanced partitioning, particularly in its min-sum version, has been studied extensively during the last two decades, with impressive results and connections to several fields of mathematics, see e.g. [LR99, ENRS99, LLR95, AR98, ARV08, KNS09, LN06, CKN09]. The min-max variants, in contrast, have received much less attention. Previously, no approximation algorithm for the Min–Max k–Partitioning problem was given explicitly, and the approximation that follows from known results is not smaller than $O(k\sqrt{\log n})$.[1] We improve this dependence on k significantly.

An important tool in our result above is an approximation algorithm for the Small-Set Expansion (SSE) problem. This problem was suggested recently by Raghavendra and Steurer [RS10] (see also [RST10a, RST10b]) in the context of the unique games conjecture. Recall that the *edge-expansion* of a subset $S \subseteq V$ with $0 < |S| \le \frac{1}{2}|V|$ is

$$\Phi(S) := \frac{\delta(S)}{|S|}.$$

The input to the SSE problem is an edge-weighted graph and $\rho \in (0, \frac{1}{2}]$, and the goal is to compute

$$\Phi_\rho := \min_{|S| \le \rho n} \Phi(S).$$

Raghavendra, Steurer and Tetali [RST10a] designed for SSE an algorithm that approximates the expansion within $O(\sqrt{(1/\Phi_\rho) \log(1/\rho)})$ factor of the optimum, while violating the bound on $|S|$ by no more than a constant factor (namely, a bicriteria approximation). Notice that the approximation factor depends on Φ_ρ; this is not an issue if every small set expands well, but in general Φ_ρ can be as small as $1/\text{poly}(n)$, in which case this guarantee is quite weak.

One can achieve a true approximation of $O(\log n)$ for SSE using [Räc08], for any value of ρ.[2] If one desires a better approximation, then an approximation of $O(\sqrt{\log n})$ using [ARV08] can be achieved at the price of slightly violating the size constraint, namely a bicriteria approximation algorithm. However, unlike the former which works for any value of ρ, the latter works only for $\rho = \Omega(1)$. In our context of min-max problems we need the case $\rho = 1/k$, where $k = k(n)$ is part of the input. Therefore, it is desirable to extend the $O(\sqrt{\log n})$ bound of [ARV08] to a large range of values for ρ.

---

[1]One could reduce the problem to the min-sum version of k-partitioning. The latter admits bicriteria approximation $O(\sqrt{\log n \log k})$ [KNS09], but the reduction loses another factor of $k/2$. Another possibility is to repeatedly remove n/k vertices from the graph, paying again a factor of $k/2$ on top of the approximation in a single iteration, which is, say, $O(\log n)$ by [Räc08].

[2]For very small values of ρ, roughly $\rho n \le O(\log^2 n)$, a better approximation ratio is known [FKN03].

1.1 Main Results

Our two main results are bicriteria approximation algorithms for the Min–Max k–Partitioning and SSE problems, presented below. The notation $O_\varepsilon(t)$ hides multiplicative factors depending on ε, i.e., stands for $O(f(\varepsilon) \cdot t)$.

Theorem 1.1. *For every positive constant $\varepsilon > 0$, Min–Max k–Partitioning admits a bicriteria approximation of $\left(O_\varepsilon(\sqrt{\log n \log k}), \ 2 + \varepsilon \right)$.*

This theorem provides a polynomial-time algorithm that with high probability outputs a partition $S_1, \ldots, S_k$ such that $\max_i |S_i| \leq (2 + \varepsilon)\frac{n}{k}$ and $\max_i \delta(S_i) \leq O(\sqrt{\log n \log k})\mathsf{OPT}$, where OPT is the optimal min-max value of partitioning into k equal-size parts. (The guarantee on part size can be improved slightly to $2 - \frac{1}{k} + \varepsilon$). This result is most interesting in the regime $1 \ll k \ll n$.

Theorem 1.2. *For every positive constant $\varepsilon > 0$, Small-Set Expansion admits a bicriteria approximation of $\left(O_\varepsilon(\sqrt{\log n \log (1/\rho)}), 1 + \varepsilon \right)$.*

This theorem provides a polynomial-time algorithm that with high probability outputs a set S of size $0 < |S| \leq (1+\varepsilon)\rho n$ whose edge-expansion is $\delta(S)/|S| = O(\sqrt{\log n \log (1/\rho)})\mathsf{OPT}$, where OPT is the minimum edge-expansion over all sets of size at most ρn. Our algorithm actually handles a more general version, called Weighted Small-Set Expansion, which is required in Theorem 1.1. We defer the precise details to Section 2.

1.2 Additional Results and Extensions

ρ–**Unbalanced Cut.** Closely related to the SSE problem is the following ρ–Unbalanced Cut problem: The input is again a graph $G = (V, E)$ with nonnegative edge-weights and a parameter $\rho \in (0, \frac{1}{2}]$, and the goal is to find a subset $S \subseteq V$ of size $|S| = \rho n$ that minimizes $\delta(S)$. The relationship between this problem and SSE is similar to the one between Balanced Cut and Sparsest Cut, and thus Theorem 1.2 yields the following result.

Theorem 1.3. *For every constant $0 < \varepsilon < 1$, the ρ–Unbalanced Cut problem admits a bicriteria approximation of $\left(O_\varepsilon(\sqrt{\log n \log(1/\rho)}), \Omega(1), 1 + \varepsilon \right)$.*

This theorem says that there is a polynomial-time algorithm that with high probability finds $S \subseteq V$ of size $\Omega(\rho n) \leq |S| \leq (1 + \varepsilon)\rho n$ and value $\delta(S) \leq O_\varepsilon(\sqrt{\log n \log (1/\rho)})\mathsf{OPT}$, where OPT is the value of an optimal solution to ρ–Unbalanced Cut. This result generalizes the bound of [ARV08] from $\rho = \Omega(1)$ to *any* value of $\rho \in (0, \frac{1}{2}]$. Our factor is better than the $O(\log n)$ true approximation ratio that follows from [Räc08], at the price of slightly violating the size constraint. Our algorithm actually handles a more general version, called Weighted ρ-Unbalanced Cut, which is required in Theorem 1.1. We defer the precise details to Section 2.4.

Min-Max-Multiway-Cut. We also consider the following Min-Max-Multiway-Cut problem, suggested by Svitkina and Tardos [ST04]: the input is an undirected graph with nonnegative edge-weights and k terminal vertices $t_1, \ldots, t_k$, the goal is to partition the vertices into k parts $S_1, \ldots, S_k$ (not necessarily balanced), under the constraint that each part contains exactly one terminal, so as to minimize $\max_i \delta(S_i)$. They designed an $O(\alpha \log n)$–approximation algorithm for this problem, where α is the approximation factor known for Minimum Bisection. Plugging $\alpha = O(\log n)$, due to Räcke [Räc08], the algorithm of Svitkina and Tardos achieves $O(\log^2 n)$-approximation. Using a similar algorithm to the one in Theorem 1.1, we obtain a better approximation factor.

Theorem 1.4. *Min-Max-Multiway-Cut admits an $O(\sqrt{\log n \log k})$–approximation algorithm.*

Somewhat surprisingly, we show that removing the dependence on n for Min-Max-Multiway-Cut (even though no balance is required) appears hard, which stands in contrast to its min-sum version, known as Multiway Cut, which admits $O(1)$–approximation [CKR00, KKS$^+$04]. The idea is to show that it would imply a similar independence of n for the min-sum version of k-partitioning, thus for large but constant k, we would get an $(O(1), O(1))$-bicriteria approximation for Min–Sum k–Partitioning, which seems unlikely based on the current state of art [ARV08, AR06, KNS09].

Theorem 1.5. *If there is a $k^{1-\varepsilon}$–approximation algorithm for Min-Max-Multiway-Cut for some constant $\varepsilon > 0$, then there is a (k^2, γ) bicriteria approximation algorithm for Min–Sum k–Partitioning with $\gamma \leq 3^{2/\varepsilon}$.*

Additionally, we also consider a common generalization of Min–Max k–Partitioning and Min-Max-Multiway-Cut, which we call Min–Max Cut. In fact we obtain Theorem 1.4 as a special case of our result for Min–Max Cut.

Excluded-minor graphs. Finally, we obtain an improved approximation – *constant factor* – for SSE in graphs excluding a fixed minor.

Theorem 1.6. *For every constant $\varepsilon > 0$, Small-Set Expansion admits:*
  - *bicriteria approximation of $\left(O_\varepsilon(r^2),\ 1+\varepsilon\right)$ on graphs excluding a $K_{r,r}$-minor.*
  - *bicriteria approximation of $\left(O_\varepsilon(\log g),\ 1+\varepsilon\right)$ on graphs of genus $g \geq 1$.*

These bounds extend to the ρ–Unbalanced Cut problem, and by plugging them into the proof of Theorems 1.1 and 1.4, we achieve an improved approximation ratio of $O(r^2)$ for Min–Max k–Partitioning and Min-Max-Multiway-Cut in graphs excluding a $K_{r,r}$-minor.

1.3 Techniques

For clarity, we restrict the discussion here mostly to our main application, Min–Max k–Partitioning. Our approach has two main ingredients. First, we reduce the problem to a weighted version of SSE, showing that an α (bicriteria) approximation for the latter can be used to achieve $O(\alpha)$ (bicriteria) approximation for Min–Max k–Partitioning. Second, we design an $O_\varepsilon(\sqrt{\log n \log(1/\rho)})$ (bicriteria) approximation for weighted SSE (recall that in our applications $\rho = 1/k$).

Let us first examine SSE, and assume for simplicity of presentation that $\rho = 1/k$. Note that SSE bears obvious similarity to both Balanced Cut and min-sum k–partition (its solution contains a single cut with a size condition, as in Balanced Cut, but the size of this cut is n/k similarly to the k pieces in min-sum k–partition). Thus, our algorithm is inspired by, but different from, the approximation algorithms known for these two problems [ARV08, KNS09]. As in these two problems, we use a semidefinite programming (SDP) relaxation to compute an ℓ_2^2 metric on the graph vertices. However, new spreading constraints are needed since SSE is highly asymmetric in its nature — it contains only a *single* cut of size n/k. We devise a randomized rounding procedure based on the *orthogonal separator* mechanics, first introduced by Chlamtac, Makarychev, and Makarychev [CMM06] in the context of unique games. These ideas lead to an algorithm that computes a cut S of *expected* size $|S| \leq O(n/k)$ and of *expected* cost $\delta(S) \leq O(\sqrt{\log n \log k})$ times the SDP value. An obvious concern is that both properties occur in only expectation and might be badly correlated,

e.g., the expected edge-expansion $\mathbb{E}[\delta(S)/|S|]$ might be extremely large. Nevertheless, we prove that with good probability, $|S| = O(n/k)$ and $\delta(S)/|S|$ is sufficiently small.

For SSE on excluded-minor and bounded-genus graphs, we give a better approximation guarantees, of a constant factor, by extending the notion of orthogonal separators to linear programs (LPs) and designing such low-distortion "LP separators" for these special graph families. The proof uses the probabilistic decompositions of Klein, Plotkin, and Rao [KPR93] and Lee and Sidiropoulos [LS10]. We believe that this result may be of independent interest. Let us note that the LP formulation for SSE is not trivial and requires novel spreading constraints. We remark that even on planar graphs, the decomposition of Räcke [Räc08] suffers an $\Omega(\log n)$ loss in the approximation guarantee, and thus does not yield $o(\log n)$ ratio for SSE on this class of graphs.

Several natural approaches for designing an approximation algorithm for Min–Max k–Partitioning fail. First, reducing the problem to trees à la Räcke [Räc08] is not very effective, because there might not be a single tree in the distribution that preserves *all* the k cuts simultaneously. Standard arguments show that the loss might be a factor of $O(k \log n)$ in the case of k different cuts. Second, one can try and formulate a relaxation for the problem. However, the natural linear and semidefinite relaxations both have large integrality gaps. As a case study, consider for a moment Min-Max-Multiway-Cut. The standard linear relaxation of Calinescu, Karloff and Rabani [CKR00] was shown by Svitkina and Tardos [ST04] to have an integrality gap of $k/2$. In Appendix A we extend this gap to the semidefinite relaxation that includes all ℓ_2^2 triangle inequality constraints. A third attempt is to repeatedly remove from the graph, using SSE, pieces of size $\Theta(n/k)$. However, by removing the "wrong" vertices from the graph, this process might arrive at a subgraph where every cut of $\Theta(n/k)$ vertices has edge-weight greater by a factor of $\Theta(k)$ than the original optimum (see Appendix B for details). Thus, a different approach is needed.

Our approach is to use multiplicative weight-updates on top of the algorithm for weighted SSE. This yields a collection $\mathcal{S}$ of sets S, all of size $|S| = \Theta(n/k)$ and cost $\delta(S) \leq O(\sqrt{\log n \log k})\mathsf{OPT}$, that covers every vertex $v \in V$ at least $\Omega(n/k)$ times. (Alternatively, this collection $\mathcal{S}$ can be viewed as a fractional solution to a configuration LP of exponential size.) Next, we randomly sample sets $S_1, \ldots, S_t$ from $\mathcal{S}$ till V is covered, and derive a partition given by $P_1 = S_1$, $P_2 = S_2 \setminus S_1$, and in general $P_i = S_i \setminus (\cup_{j<i} S_j)$. This step is somewhat counter-intuitive, since the sets P_i may have very large cost $\delta(P_i)$ (because a set P_i might be a strict subset of a set $S_{i'}$). We show that the total expected boundary of the partition is not very large, i.e., $\mathbb{E}[\sum_i \delta(P_i)] \leq O(k\sqrt{\log n \log k})\mathsf{OPT}$. Then, we start fixing the partition by the following local operation: find a P_i violating the constraint $\delta(P_i) \leq O(\sqrt{\log n \log k})\mathsf{OPT}$, replace it with the unique S_i containing it, and adjust other sets P_j accordingly. Somewhat surprisingly, we prove that this local fixing procedure terminates (quickly). Finally, the resulting partition consists of sets P_i, each of which satisfies the necessary properties, but now the number of these sets might be very large. So the last step is to merge small sets together. We show that this can be done while maintaining simultaneously the constraints on the sizes and on the costs of the sets.

Organization. We first show in Section 2 how to approximate Weighted Small-Set Expansion (in both general and excluded-minor graphs). We then show in Section 2.4 that an approximation algorithm for Weighted Small-Set Expansion also yields one for Weighted ρ-Unbalanced Cut. In Section 3 we present an approximation algorithm for Min–Max k–Partitioning that uses the aforementioned algorithm for ρ–Unbalanced Cut (and in turn the one for Weighted Small-Set Expansion). The common generalization of both Min–Max k–Partitioning and Min-Max-Multiway-Cut, Min–Max

Cut, appears in Section 4. Theorem 1.5 is proved in Section 5.

2 Approximation Algorithms for Small Set Expansion

In this section we design approximation algorithms for the Small-Set Expansion problem. Our main result is for general graphs and uses an SDP relaxation. It actually holds for a slight generalization of the problem, where expansion is measured with respect to vertex weights (see Definition 2.1 and Theorem 2.1). We further obtain improved approximation for certain graph families such as planar graphs (see Section 2.3).

To simplify notation, we shall assume that vertex weights are normalized: we consider measures μ and η with $\mu(V) = \eta(V) = 1$. We denote $\mu(u) = \mu(\{u\})$ and $\eta(u) = \eta(\{u\})$. We let (V, w) denote a complete (undirected) graph on vertex set V with edge-weight $w(u,v) = w(v,u) \geq 0$ for every $u \neq v \in V$. In our context, such (V, w) can easily model a specific edge set E, by simply setting $w(u, v) = 0$ for every non-edge $(u, v) \notin E$. Recall that we let $\delta(S) := \sum_{u \in S, v \in V \setminus S} w(u, v)$ be the total weight of edges crossing the cut $(S, V \setminus S)$, and further let $w(E)$ denote the total weight of all edges.

Definition 2.1 (Weighted Small-Set Expansion). *Let $G = (V, w)$ be a graph with nonnegative edge-weights, and let μ and η be two measures on the vertex set V with $\mu(V) = \eta(V) = 1$. The weighted small set expansion with respect to $\rho \in (0, 1/2]$ is*

$$\Phi_{\rho,\mu,\eta}(G) \quad := \quad \min\left\{\frac{\delta(S)}{w(E)} \times \frac{1}{\eta(S)} \; : \; \eta(S) > 0, \; \mu(S) \leq \rho\right\}.$$

Theorem 2.1 (Approximating SSE). *(I) For every fixed $\varepsilon > 0$, there is a polynomial-time algorithm that given as input an edge-weighted graph $G = (V, w)$, two measures μ and η on V ($\mu(V) = \eta(V) = 1$), and some $\rho \in (0, 1/2]$, finds a set $S \subset V$ satisfying $\eta(S) > 0$, $\mu(S) \leq (1 + \varepsilon)\rho$ and*

$$\frac{\delta(S)}{w(E)} \times \frac{1}{\eta(S)} \quad \leq \quad D \times \Phi_{\rho,\mu,\eta}(G), \tag{1}$$

where $D = O_\varepsilon(\sqrt{\log n \log(1/\rho)})$.
(II) When the input contains in addition a parameter $H \in (0, 1)$, the algorithm finds a non-empty set $S \subset V$ satisfying $\mu(S) \leq (1 + \varepsilon)\rho$, $\eta(S) \in [\Omega(H), 2(1 + \varepsilon)H]$, and

$$\frac{\delta(S)}{w(E)} \times \frac{1}{\eta(S)} \quad \leq \quad D \times \min\left\{\frac{\delta(S)}{w(E)} \times \frac{1}{\eta(S)} \; : \; \eta(S) \in [H, 2H], \; \mu(S) \leq \rho\right\}, \tag{2}$$

where $D = O_\varepsilon(\sqrt{\log n \log(\max\{1/\rho, 1/H\})})$.

We prove part I of the theorem in Section 2.1, and part II in Section 2.2. These algorithms require the following notion of m-orthogonal separators due to Chlamtac, Makarychev, and Makarychev [CMM06].

Definition 2.2 (Orthogonal Separators). *Let X be an ℓ_2^2 space (i.e., a collection of vectors satisfying ℓ_2^2 triangle inequalities). We say that a distribution over subsets of X is an m-orthogonal separator of X with distortion D, probability scale $\alpha > 0$ and separation threshold $\beta < 1$ if the following conditions hold for $S \subset X$ chosen according to this distribution:*

- *For all $u \in X$ we have $\Pr(u \in S) = \alpha \|u\|^2$.*
- *For all $u, v \in X$ with $\|u - v\|^2 \geq \beta \min(\|u\|^2, \|v\|^2)$,*

$$\Pr(u \in S \text{ and } v \in S) \quad \leq \quad \frac{\min\{\Pr(u \in S), \Pr(v \in S)\}}{m}.$$

- *For all $u, v \in X$ we have $\Pr(I_S(u) \neq I_S(v)) \leq \alpha D \times \|u - v\|^2$, where I_S is the indicator function of the set S.*

Theorem 2.2 ([CMM06]). *There exists a polynomial-time randomized algorithm that given a set of vectors X, positive number m, and $\beta < 1$ generates m-orthogonal separator with distortion $D = O_\beta(\sqrt{\log|X|\log m})$ and scale $\alpha \geq 1/p(|X|)$ for some polynomial p.*

In the original paper [CMM06], the second requirement in the definition of orthogonal separators was slightly different, however, exactly the same algorithm and proof works in our case: If $\|u-v\|^2 \geq \beta\|u\|^2$ and $\|u\|^2 \leq \|v\|^2$, then $\langle u, v \rangle = (\|u\|^2 + \|v\|^2 - \|u-v\|^2)/2 \leq ((1-\beta)\|u\|^2 + \|v\|^2)/2 \leq (1 - \beta/2)\|v\|^2$. Then, by Lemma 4.1 in [CMM06], $\langle \varphi(u), \varphi(v) \rangle \leq (1-\beta/2)$; hence $\|\varphi(u) - \varphi(v)\|^2 \geq \beta > 0$ and, in Corollary 4.6, $\|\psi(u) - \psi(v)\| \geq 2\gamma = \sqrt{\beta}/4 > 0$.

2.1 Algorithm I: Small-Set Expansion in General Graphs

We now prove part I of Theorem 2.1.

SDP Relaxation. In our relaxation we introduce a vector $\bar{v}$ for every vertex $v \in V$. In the intended solution of the SDP corresponding to the optimal solution $S \subset V$, $\bar{v} = 1$ (or, a fixed unit vector e), if $v \in S$; and $\bar{v} = 0$, otherwise. The objective is to minimize the fraction of cut edges

$$\min \quad \frac{1}{w(E)} \sum_{(u,v) \in E} w(u,v) \|\bar{u} - \bar{v}\|^2.$$

We could constrain all vectors $\bar{v}$ to have length at most 1, i.e. $\|\bar{v}\|^2 \leq 1$, but it turns out our algorithm never uses this constraint. We require that the vectors $\{\bar{v} : v \in V\} \cup \{0\}$ satisfy ℓ_2^2 triangle inequalities i.e., for every $u, v, w \in V$, $\|\bar{u}-\bar{w}\|^2 \leq \|\bar{u}-\bar{v}\|^2 + \|\bar{v}-\bar{w}\|^2$, $\|\bar{u}\|^2 \leq \|\bar{u}-\bar{v}\|^2 + \|\bar{v}\|^2$, $\|\bar{u} - \bar{w}\|^2 \leq \|\bar{u}\|^2 + \|\bar{w}\|^2$. Suppose now that we have approximately guessed the measure H of the optimal solution $H \leq \eta(S) \leq 2H$ (this step is not necessary but it simplifies the exposition; in fact, we could simply let $H = 1$, since the SDP is otherwise homogeneous). This can be done since the measure of every set S lies in the range from $\eta(u)$ to $n\eta(u)$, where u is the heaviest element in S, hence H can be chosen from the set $\{2^t\eta(u) : u \in V, t = 0, \cdots, \lfloor \log_2 n \rfloor\}$ of size $O(n \log n)$. Then we add a constraint

$$\sum_{v \in V} \|\bar{v}\|^2 \eta(v) \quad \geq \quad H. \tag{3}$$

We denote $\eta(u) = \eta(\{u\})$ and $\mu(u) = \mu(\{u\})$. Finally, we introduce new spreading constraints: for every $u \in V$,

$$\sum_{v \in V} \mu(v) \cdot \min\{\|\bar{u} - \bar{v}\|^2, \|u\|^2\} \quad \geq \quad (1 - \rho)\|\bar{u}\|^2.$$

(Alternatively, we could use a slightly simpler, almost equivalent constraint $\sum_{v \in V} \langle \bar{u}, \bar{v} \rangle \mu(v) \leq \rho\|\bar{u}\|^2$. We chose to use the former formulation because an analogous constraint can be written in a

linear program, see Section 2.3.) In the intended solution this constraint is satisfied, since if $u \in S$, then $\bar{u} = 1$ and the sum above equals $\mu(V \setminus S) \geq 1 - \rho$. If $u \notin S$, then $\bar{u} = 0$ and both sides of the constraint equal 0.

The SDP relaxation used in our algorithm is presented below in its entirety. Note that the second constraint can be written as $\langle u, v \rangle \leq \|u\|^2$, and the third constraint can be written as $\langle u, v \rangle \geq 0$.

$$
\begin{aligned}
\textbf{min} \qquad & \frac{1}{w(E)} \sum_{(u,v) \in E} w(u,v) \, \|\bar{u} - \bar{v}\|^2 \\
\textbf{s.t.} \quad \|\bar{u} - \bar{w}\|^2 + \|\bar{w} - \bar{v}\|^2 \ & \geq \ \|\bar{u} - \bar{v}\|^2, \qquad \forall u,v,w \in V, \\
\|\bar{u} - \bar{v}\|^2 \ & \geq \ \|\bar{u}\|^2 - \|\bar{v}\|^2, \quad \forall u,v \in V, \\
\|\bar{u}\|^2 + \|\bar{v}\|^2 \ & \geq \ \|\bar{u} - \bar{v}\|^2, \qquad \forall u,v \in V, \\
\sum_{v \in V} \mu(v) \cdot \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\} \ & \geq \ (1 - \rho)\|\bar{u}\|^2, \quad \forall \, u \in V, \\
\sum_{v \in V} \|\bar{v}\|^2 \eta(v) \ & \geq \ H.
\end{aligned}
\qquad (4)
$$

We now describe the approximation algorithm.

Approximation Algorithm. We first informally describe the main idea behind the algorithm. The algorithm solves the SDP relaxation and obtains a set of vectors $\{\bar{u}\}_{u \in V}$. Now it samples an orthogonal separator, a random set $S \subset V$, and returns it. Assume for the moment that $\alpha = 1$. Since $\Pr(v \in S) = \|\bar{v}\|^2$, we get $\mathbb{E}[\eta(S)] \geq H$. The expected size of the cut is at most $D \times SDP$ by the third property of orthogonal separators; and thus the sparsity is at most $D \times SDP/H \leq 2D \times OPT$. The second property of orthogonal separators guarantees that if $\bar{u} \in S$, then the vectors that are far from $\bar{u}$, a very small fraction will belong to S (since the conditional probability $\Pr(\bar{v} \in S \mid \bar{u} \in S) \leq 1/m$ is very small). And by the spreading constraints, at most $(1 + \varepsilon)\rho$ fraction of vectors (w.r.t. the measure μ) is close to $\bar{u}$. Hence, the total expected measure of S is at most $(1 + \varepsilon)\rho + 1/m \leq (1 + 2\varepsilon)\rho$. We now proceed to the formal argument.

We may assume that ε is sufficiently small i.e., $\varepsilon \in (0, 1/4)$. The approximation algorithm guesses approximate value of the weight H: $H \leq \eta(S) \leq 2H$. Set the length of all vectors $\bar{u}$ with $\eta(u) > 2H$ to be 0. Solves the SDP and obtains a set of vectors $X = \{\bar{v}\}_{v \in V}$. Then, it finds an orthogonal separator S with $m = \max(\varepsilon^{-1}\rho^{-1})$ and $\beta = \varepsilon$. For convenience, we let S be the set of vertices corresponding to vectors belonging to the orthogonal separator rather than the vectors themselves. The algorithm repeats the previous step $\lceil \alpha^{-1} n^2 \rceil$ times (recall α is the probabilistic scale of the orthogonal separator) and outputs the best S satisfying $0 < \mu(S) < (1 + 10\varepsilon)\rho$. With an exponentially small probability no S satisfies this constraint, in which case, the algorithm outputs an arbitrary set satisfying constraints.

Analysis. We first estimate the probability of the event "$u \in S$ and $\mu(S) < (1 + 10\varepsilon)\rho$" for a fixed vertex $u \in V$. Let $A_u = \{v : \|\bar{u} - \bar{v}\|^2 \geq \beta\|\bar{u}\|^2\}$ and $B_u = \{v : \|\bar{u} - \bar{v}\|^2 < \beta\|\bar{u}\|^2\}$. We show that only a small fraction of A_u belongs to S, and that the set B_u is small.

From the spreading constraint $\sum_{v \in V} \min(\|\bar{u} - \bar{v}\|^2, \|u\|^2)\mu(v) \geq (1 - \rho)\|\bar{u}\|^2$, by the Markov inequality, we get that $\mu(B_u) \leq \rho/(1 - \beta) \leq (1 + 2\varepsilon)\rho$. For an arbitrary $v \in A_u$ (for which $\bar{v} \neq 0$) write $\|\bar{u} - \bar{v}\|^2 \geq \beta\|\bar{u}\|^2 \geq \beta \min(\|\bar{u}\|^2, \|\bar{v}\|^2)$. By the second property of orthogonal separators, $\Pr(v \in S \mid u \in S) \leq 1/m$, thus the expected measure $\mu(A_u \cap S)$ is at most $\mathbb{E}\mu(A_u \cap S) \leq \varepsilon\rho$. Now, by the Markov inequality, given that $u \in S$, the probability of the bad event "$\mu(S) \geq (1 + 10\varepsilon)\rho$ (and, thus $\mu(A_u \cap S) \geq 8\varepsilon\rho$)" is at most $1/8$. Each vertex $u \in V$ belongs to S with probability $\alpha\|\bar{u}\|^2$. Hence, $u \in S$, and $\mu(S) < (1 + 10\varepsilon)\rho$ with probability at least $3/4\,\alpha\|\bar{u}\|^2$.

Finally, we use the third property of orthogonal separators to bound the size of the cut $\delta(S)$

$$\mathbb{E}\delta(S) = \sum_{(u,v) \in E} |I_S(u) - I_S(v)|w(u,v) \leq \alpha D \times \sum_{(u,v) \in E} \|\bar{u} - \bar{v}\|^2 w(u,v) = \alpha D \times SDP \times w(E).$$

Here, as usual, SDP denotes the value of the SDP solution; and $D = O_\varepsilon(\sqrt{\log n \log(1/\delta)})$ is the distortion of m-orthogonal separators.

Define function

$$f(S) = \eta(S) - \frac{\delta(S)}{w(E)} \times \frac{H}{4D \times SDP},$$

if $|S| \neq \varnothing$, $\mu(S) < (1 + 10\varepsilon)\rho$, and $f(S) = 0$, otherwise. The expectation

$$\mathbb{E}f(S) \geq \sum_{u \in V} \frac{3\alpha\|\bar{u}\|^2\eta(u)}{4} - \frac{\alpha H}{4} \geq \frac{\alpha H}{2}.$$

The random variable $f(S)$ is always bounded by $2nH$, thus with probability at least α/n, $f(S) > 0$. Therefore, with probability exponentially close to 1, after $\alpha^{-1}n^2$ iterations, the algorithm will find S with $f(S) > 0$. Since $f(S) > 0$, we get $\eta(S) > 0$, $\mu(S) < (1 + 10\varepsilon)\rho$, and

$$\frac{\delta(S)}{w(E)} \times \frac{1}{\eta(S)} \leq 4D \times \frac{SDP}{H}.$$

This finishes the proof of part I since $SDP/(2H) \leq \Phi_{\rho,\mu,\eta}(G)$.

2.2 Algorithm II: Small-Set Expansion in General Graphs

We now prove part II of Theorem 2.1. This algorithm uses an SDP relaxation similar to part I, although we need a few additional constraints. We write a constraint ensuring that "$\eta(S) \leq 2H$" (recall H is an approximate value of $\eta(S)$ in the optimal solution): we add spreading constraints for all $u \in V$,

$$\sum_{v \in V} \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\}\,\eta(v) \leq 2H\|\bar{u}\|^2,$$

and we let $m = \max\{\varepsilon^{-1}\rho^{-1}, H^{-1}\rho^{-1}\}$. We also require

$$\sum_{v \in V} \|v\|^2\mu(v) \leq \rho. \tag{5}$$

Algorithm II gets H, the approximate value of the measure $\eta(S)$, as input, and thus does not need to guess it.

9

Remark 2.3. *To handle terminals in the extended version of the problem (see Section 4) we guess which terminal $u \in T$ belongs to the optimal solution S (if any), and set $\|\bar{u}\| = 1$ and $\|\bar{v}\| = 0$ for $v \in T \setminus \{u\}$. Since an orthogonal separator never contains the zero vector, we will never choose more than one terminal in the set S.*

Approximation Algorithm. The algorithm consists of many iterations of a slightly modified Algorithm I. At every step the algorithm obtains a set S of vertices (returned by Algorithm I) and adds it to the set T, which is initially empty. Then, the algorithm removes vectors corresponding to S from the set X, the SDP solution, and repeats the same procedure till $\mu(T) \geq \rho/4$ or $\eta(T) \geq H/4$. In the end, the algorithm returns the set T if $\mu(T) \leq \rho$ and $\eta(T) \leq H$, and the last set S otherwise.

The algorithm changes the SDP solution (by removing some vectors), however we can ignore these changes, since the objective value of the SDP may only decrease and all constraints but (3) are still satisfied. Since the total weight $\eta(T)$ of removed vertices is at most $H/4$, a slightly weaker variant of constraint (3) is satisfied. Namely,

$$\sum_{u \in V} \|\bar{u}\|^2 \eta(u) \geq 3H/4. \tag{3'}$$

We now describe the changes in Algorithm I: instead of f, we define function f':

$$f'(S) = \eta(S) - \frac{\delta(S)}{w(E)} \times \frac{H}{4D \times SDP} - \frac{\mu(S)}{4\rho} \times H,$$

if $|S| \neq \varnothing$, $\mu(S) < (1+10\varepsilon)\rho$ and $\eta(S) \leq (1+10\varepsilon)H$ and $f'(S) = 0$, otherwise. Notice, that f' has an extra term comparing to f and, in order for $f'(S)$ to be positive, the constraint $\eta(S) \leq 2(1+10\varepsilon)H$ should be satisfied. The new variant of Algorithm I, returns S, once $f'(S) > 0$.

The same argument as before shows that for any given $u \in V$ conditional on $u \in S$, $\mu(S) \leq (1+10\varepsilon)\rho$ and $\eta(S) \leq 2(1+10\varepsilon)H$ with probability at least $3/4$. Then, using a new constraint (5), we get $\mathbb{E}\mu(S) \leq \alpha\rho$. Hence, the expectation

$$\mathbb{E}f'(S) \geq \frac{3\alpha \times 3/4\,H}{4} - \frac{\alpha H}{4} - \frac{\alpha H}{4} \geq \frac{\alpha H}{16}.$$

Again, after at most $O(\alpha^{-1}n^2)$ iterations the algorithm will find S with $f'(S) > 0$ (and only with exponentially small probability fail)[3]. Then, $f'(S) > 0$ implies

$$\frac{\delta(S)}{w(E)} \leq 4D \times \frac{SDP}{H}\eta(S); \tag{6}$$

and $\eta(S) \geq H \times \mu(S)/(4\rho)$.

The last inequality implies that at every moment $\eta(T) \geq H \times \mu(T)/(4\rho)$. Hence, if $\mu(T) \geq \rho/4$ (recall, this is one of the two conditions, when the algorithm stops), then $\eta(T) \geq H/16$. Therefore, if the algorithm returns set T, then $\eta(T) \geq H/16$. If the algorithm returns set S then either $\mu(S) \geq 3/4\,\rho$ and thus $\eta(S) \geq 3H/16$ or $\eta(S) \geq 3/4\,H$.

Both, $\mu(T)$ and $\eta(T)$ are bounded from above by ρ and H respectively; $\mu(S)$ and $\eta(S)$ are bounded from above by $(1+10\varepsilon)\rho$ and $2(1+10\varepsilon)H$ respectively.

The inequality (6) holds for every set S added in T, hence this inequality holds for T.

---

[3]In fact, now $f'(S) \leq 2H$, thus we need only $O(\alpha^{-1}n)$ iterations.

2.3 Small-Set Expansion in Minor-Closed Graph Families

In this subsection we prove Theorem 1.6. We start by writing an LP relaxation. For every vertex $u \in V$ we introduce a variable $x(u)$ taking values in $[0,1]$; and for every pair of vertices $u, v \in V$ we introduce a variable $z(u,v) = z(v,u)$ also taking values in $[0,1]$. In the intended integral solution corresponding to a set $S \subset V$, $x(u) = 1$ if $u \in S$, and $x(u) = 0$ otherwise; $z(u,v) = |x(u) - x(v)|$. (One way of thinking of $x(u)$ is as the distance to some imaginary vertex O that never belongs to S. In the SDP relaxation vertex O is the origin.) It is instructive to think of $x(u)$ as an analog of $\|\bar{u}\|^2$ and of $z(u,v)$ as an analog of $\|\bar{u} - \bar{v}\|^2$.

It is easy to verify that LP (7) below is a relaxation of the Small-Set Expansion problem. It has a constraint saying that $z(u,v)$ is a metric (or, strictly speaking, semi-metric). A novelty of the LP is in the third constraint, which is a new spreading constraints for ensuring the size of S is small.

$$
\begin{aligned}
\min \quad & \frac{1}{w(E)} \sum_{(u,v) \in E} w(u,v)\, \|\bar{u} - \bar{v}\|^2 \\[2mm]
\text{s.t.} \quad z(u,v) + z(v,w) &\geq z(u,w), && \forall u,v,w \in V, \\
|x(u) - x(v)| &\leq z(u,v), && \forall u,v \in V, \\
\sum_{v \in V} \mu(v) \cdot \min x(u), z(u,v) &\geq (1-\rho)x(u), && \forall u \in V, \\
x(u), z(u,v) &\in [0,1], && \forall u,v \in V.
\end{aligned}
$$

$$(7)$$

We introduce an analog of m-orthogonal separators for linear programming, which we call LP separators.

Definition 2.4 (LP separator). *Let $G = (V,E)$ be a graph, and let $\{x(u), z(u,v)\}_{u,v \in V}$ be a set of numbers. We say that a distribution over subsets of V is an LP separator of V with distortion $D \geq 1$, probability scale $\alpha > 0$ and separation threshold $\beta \in (0,1)$ if the following conditions hold for $S \subset V$ chosen according to this distribution:*
- *For all $u \in V$, $\Pr(u \in S) = \alpha\, x(u)$.*
- *For all $u,v \in V$ with $z(u,v) \geq \beta \min\{x(u), x(v)\}$, $\Pr(u \in S \text{ and } v \in S) = 0$.*
- *For all $(u,v) \in E$, $\Pr(I_S(u) \neq I_S(v)) \leq \alpha D \times z(u,v)$, where I_S is an indicator for the set S.*

Below we present an efficient algorithm for an LP separator: given a graph $G = (V,E)$ excluding $K_{r,r}$ as a minor, a parameter $\beta \in (0,1)$, and a set of numbers $\{x(u), z(u,v)\}_{u,v \in V}$ satisfying the triangle inequalities described above (but not necessarily the spreading constraints), the algorithm computes an LP separator with distortion $O(r^2)$ (for genus g graphs the distortion is $O(\log g)$). This proves Theorem 1.6 as follows: by replacing in the algorithms above the SDP relaxation (4) with the LP relaxation (7), and the orthogonal separators with LP separators, we obtain $O(r^2)$ approximation algorithm approximation algorithm for SSE in $K_{r,r}$ excluded-minor graphs. Combined with the framework in Section 3, we consequently obtain an $O(r^2)$-approximation algorithm for Min–Max k–Partitioning and Min-Max-Multiway-Cut on such graphs.

Computing LP Separators. We now describe an algorithm that samples an LP separator (see Definition 2.4) with respect to a feasible solution to LP (7). We recall a standard notion of low-diameter decomposition of a metric space, see e.g. [Bar96, GKL03, KR11] and references therein.

Let (V, d) be a finite metric space. Given a partition P of V and a point $v \in V$, we refer to the elements of P as clusters, and let $P(v)$ denote the cluster $S \in P$ that contains v, so $v \in S \in \mathcal{P}$. A *stochastic decomposition* of this metric is a probability distribution ν over partitions P of V.

Definition 2.5 (Separating Decomposition). *Let $D, \Delta > 0$. A stochastic decomposition ν of a finite metric space (V, d) is called a D-separating Δ-bounded decomposition if it satisfies:*

- *For every partition $P \in \operatorname{supp}(\nu)$ and every cluster $S \in P$,*

$$\operatorname{diam}(S) := \max_{u,v \in S} d(u, v) \leq \Delta.$$

- *For every $u, v \in V$, the probability that a partition P sampled from ν separates them is*

$$\Pr_{P \sim \nu}[P(u) \neq P(v)] \leq D \cdot \frac{d(u, v)}{\Delta}.$$

Theorem 2.3 ([KPR93, Rao99, FT03]). *Let $G = (V, E)$ be a graph excluding $K_{r,r}$ as a minor, equipped with nonnegative edge-lengths. Then the graph's shortest-path metric d_G admits, for every $\Delta > 0$, an $O(r^2)$-separating Δ-bounded decomposition. Moreover, there is a polynomial-time algorithm that samples a partition from this distribution.*

Lee and Sidiropoulos [LS10] show similarly for graphs with genus $g \geq 1$ an $O(\log g)$-separating decomposition. Alternative algorithms for both cases are shown in [KR11].

Definition 2.6 (Probabilistic Partitioning). *Consider a graph $G = (V, E)$ and nonnegative numbers $\{x(u), z(u, v)\}_{u,v \in V}$. We say that a distribution ν over partitions of V is called a probabilistic partitioning with distortion $D > 0$ and separation threshold $\beta > 0$ if the following properties hold:*

- *For every edge $(u, v) \in E$ with $x(u) > 0$:*

$$\Pr_{P \sim \nu}(P(u) \neq P(v)) \leq D \cdot z(u, v)/x(u).$$

- *For every $u, v \in V$ with $z(u, v) \geq \beta x(u)$, we have $P(u) \neq P(v)$ for all $P \in \operatorname{supp}(\nu)$.*

Theorem 2.4 (Separating decomposition implies probabilistic partitioning). *Let $G = (V, E)$ be a graph that excludes $K_{r,r}$ as a minor, and let $\{x(u), z(u, v)\}_{u,v \in V}$ satisfy the first two constraints of LP (7). Then for every $\beta \in (0, 1]$, there is a probabilistic partitioning ν with distortion $D = O(r^2 \beta^{-1})$ and separation threshold β.*

Proof. We define new lengths $y(u, v) = \min\{1, z(u, v)/x(u), z(u, v)/x(v)\}$ for all $u, v \in V$; by convention, if $x(u) = 0$ or $x(v) = 0$ then define $y(u, v) = 0$. (We remark that a similar approach was used in [CKR00]). These lengths may violate the triangle inequality. Let $d : V \times V \to \mathbb{R}$ be the shortest path metric in graph G with edge lengths equal $y(u, v)$ (note: $y(u, v)$ is defined for all pairs $u, v \in V$; however, to obtain d we look only at $(u, v) \in E$). Clearly, for every edge $(u, v) \in E$, $d(u, v) \leq y(u, v)$. On the other hand, as we show below, for every $u, v \in V$, $d(u, v) \geq \frac{2}{3} y(u, v)$.

Claim 2.5. *For all $u, v \in V$ we have $d(u, v) \geq \frac{2}{3} y(u, v)$.*

Proof. Pick two vertices $u, v \in V$ and consider an arbitrary path $u = w_1, w_2, \ldots, w_N = v$. We prove that the length of the path (in which the length of each edge (w_i, w_{i+1}) is $y(w_i, w_{i+1})$) is at least $\frac{2}{3} y(u, v)$. If the length of the path is greater than $2/3$ we are done. Thus, we may assume that the lengths of all edges are at most $2/3 < 1$. We also assume that $x(u) \geq x(v)$ and thus $y(u, v) \leq z(u, v)/x(u)$. We have,

$$
\begin{aligned}
\sum_{i=1}^{N-1} y(w_i, w_{i+1}) &= \sum_{i=1}^{N-1} \frac{z(w_i, w_{i+1})}{\max(x(w_i), x(w_{i+1}))} \geq \frac{1}{\max_i(x(w_i))} \sum_{i=1}^{N-1} z(w_i, w_{i+1}) \\
&\geq \frac{z(u, v)}{\max_i(x(w_i))} \geq \frac{x(u)\, y(u, v)}{\max_i(x(w_i))}.
\end{aligned}
$$

The second inequality holds since $z(\cdot, \cdot)$ is a metric. If $x(u)/\max_i(x(w_i)) \geq 2/3$, we are done. Assume that for $j = \operatorname{argmax} x(w_j)$, $x(u)/x(w_j) < 2/3$. Then,

$$
\begin{aligned}
\sum_{i=1}^{N-1} y(w_i, w_{i+1}) &\geq \sum_{i=1}^{N-1} \frac{z(w_i, w_{i+1})}{x(w_j)} \geq \sum_{i=1}^{N-1} \frac{|x(w_i) - x(w_{i+1})|}{x(w_j)} \\
&\geq \frac{|x(w_j) - x(u)| + |x(w_j) - x(v)|}{x(w_j)} \geq 2\left(1 - \frac{x(u)}{x(w_j)}\right) \geq \frac{2}{3}.
\end{aligned}
$$

This finishes the proof of Claim 2.5. □

We now apply the theorem of Klein, Plotkin, and Rao [KPR93] to the metric $d(u, v)$ and obtain a probabilistic partition $\mathcal{P}$ with $\Delta = \beta/3$ and $D' = O(r^2)$. This partition satisfies the following properties.

- If $z(u, v) \geq \beta x(u)$ for $u, v \in V$, then either $x(u) \geq x(v)$ and hence $y(u, v) \geq \beta$, or $x(v) \geq x(u)$, then (using $z(u, v) \geq x(v) - x(u)$)

$$
z(u, v) \geq \frac{\beta}{2}(x(v) - x(u)) + \left(1 - \frac{\beta}{2}\right) z(u, v) \geq \frac{\beta}{2}(x(v) - x(u)) + \left(1 - \frac{\beta}{2}\right)\beta x(u) \geq \frac{\beta}{2} x(v).
$$

  Thus, $y(u, v) \geq \beta/2$ in either case and $d(u, v) \geq 2/3\, y(u, v) \geq \beta/3 \equiv \Delta$ (by Claim 2.5).

- For every $(u, v) \in E$,

$$
\Pr(P_u \neq P_v) \leq D' \frac{d(u, v)}{\Delta} \leq \frac{D'}{\Delta} \frac{z(u, v)}{x(u)}.
$$

The distortion D equals $D'/\Delta = O(r^2/\beta)$. □

Given a solution for LP (7) (the relaxation for SSE problem), we could proceed as follows: Construct a probabilistic partition $\mathcal{P}$ with distortion $D = O(r^2 \beta^{-1})$ and some constant separation threshold $\beta \in (0, 1)$, then pick a random vertex $w \in V$ with probability $x(w)\eta(w)/\sum_u x(u)\eta(u)$ and, finally, output the cluster P_w. However, to highlight the similarity between this LP-algorithm and the previous SDP-algorithm (for general graphs), we give an algorithm for constructing LP separators, which in turn is used by the Small-Set Expansion algorithm.

Theorem 2.6. *There exists an algorithm that given a graph $G = (V, E)$ with an excluded minor $K_{r,r}$, a set of numbers $\{x(u), z(u, v)\}_{u,v \in V}$ satisfying the triangle inequality constraints, and a parameter $\beta \in [0, 1]$, returns an LP separator $S \subset V$ with distortion $D = r^2 \beta^{-1}$ and separation threshold β.*

Algorithm. The algorithm samples a random partition P with distortion $D = O(r^2 \beta^{-1})$ and a separation threshold β. For every $C \in P$, let

$$x_\infty(C) = \max_{u \in C} x(u).$$

The algorithm picks a random set $S \in P$ with probability $\Pr(S = C) = x_\infty(C)/n$; and with the remaining probability

$$1 - \frac{1}{n} \sum_{C \in P} x_\infty(C) \geq 1 - \frac{1}{n} \sum_{u \in V} x(u) \geq 0,$$

the algorithm sets $S = \varnothing$.

Now, to guarantee that every vertex u is chosen with probability exactly $\alpha x(u)$, where $\alpha = 1/n$, the algorithm removes some elements from S: it picks at random $t \in [0, 1]$ and outputs set

$$S' = \{u \in S : x(u) \geq tx_\infty(S)\}.$$

Analysis. Verify that S' satisfies the properties of LP separators (with $\alpha = 1/n$). For every $u \in V$,

$$\Pr(u \in S') = \mathbb{E}_P\Big[\Pr(S = P(u) \mid P) \cdot \Pr(x(u) \geq tx_\infty(P(u))) \mid P\Big] = \mathbb{E}_P\left[\frac{x_\infty(P(u))}{n} \cdot \frac{x(u)}{x_\infty(P(u))}\right] = \frac{x(u)}{n}.$$

Then, if $z(u, v) \geq \min(x(u), x(v))$, then $P(u) \neq P(v)$ and hence

$$\Pr(u, v \in S) := \Pr(P(u) = P(v) = S) = 0.$$

Finally,

$$\Pr(u \in S', v \notin S') \leq \Pr(v \notin S \mid u \in S') \Pr(u \in S') + \Pr(x(v) \leq tx_\infty(S) \mid u \in S') \Pr(u \in S').$$

We estimate the first term (using that ν has distortion $D = O(r^2 \beta^{-1})$; see Definition 2.6)

$$\Pr(u \in S, v \notin S) = \frac{x(u)}{n} \Pr(v \notin S \mid u \in S) \leq \frac{x(u)}{n} \times D \frac{z(u, v)}{x(u)} = \frac{D}{n} z(u, v),$$

and then the second term

$$
\begin{aligned}
\Pr(x(v) \leq tx_\infty(S) \mid u \in S') \Pr(u \in S') &= \frac{x(u)}{n} \Pr(x(v) \leq tx_\infty(P_u) \mid x(u) \geq tx_\infty(P_u), S = P_u) \\
&= \frac{x(u)}{n} \Pr(t \geq x(v)/x_\infty(P_u) \mid t \leq x(u)/x_\infty(P_u)) \\
&= \frac{x(u)}{n} \times \frac{\min(0, x(u) - x(v))}{x(u)} \leq \frac{z(u, v)}{n}.
\end{aligned}
$$

This completes the proof of Theorem 2.6.

14

2.4 From SSE to ρ–Unbalanced Cut

ρ–Unbalanced Cut and SSE are equivalent, up to some constants, with respect to bicriteria approximation guarantees. Indeed, the two problems are related in the same way that Balanced Cut and Sparsest Cut are. We refer the reader to [LR99, RST10a], and omit details from this version of the paper.

Our intended application of approximating Min–Max k–Partitioning (in Section 3), requires a weighted version of the ρ–Unbalanced Cut problem, as follows.

Definition 2.7 (Weighted ρ-Unbalanced Cut)**.** *The input to this problem is a tuple $\langle G, y, w, \tau, \rho \rangle$, where $G = (V, E)$ is a graph with vertex-weights $y : V \to \mathbb{R}_+$, edge-costs $w : E \to \mathbb{R}_{\geq 0}$, and parameters $\tau, \rho \in (0, 1]$. The goal is to find $S \subseteq V$ of minimum cost $\delta(S)$ satisfying:*

1. $y(S) \geq \tau \cdot y(V)$; and

2. $|S| \leq \rho \cdot n$.

The unweighted version of the problem (defined in Section 1.2) has $\tau = \rho$ and unit vertex-weights, i.e. $y(v) = 1$ for all $v \in V$. We focus on the direction of reducing Weighted ρ-Unbalanced Cut to Weighted Small-Set Expansion, which is needed for our intended application. Formally, we have the following corollary of Theorem 2.1. We use $\mathsf{OPT}_{\langle G, y, w, \tau, \rho \rangle}$ to denote the optimal value of the corresponding weighted ρ–Unbalanced Cut instance.

Corollary 2.7 (Approximating ρ-Unbalanced Cut)**.** *For every $\varepsilon > 0$, there exists a polynomial-time algorithm that given an instance $\langle G, y, w, \tau, \rho \rangle$ of Weighted ρ-Unbalanced Cut, finds a set S satisfying $|S| \leq \beta \rho n$, $y(S) \geq \tau/\gamma$ and $\delta(S) \leq \alpha \cdot \mathsf{OPT}_{\langle G, y, w, \tau, \rho \rangle}$ for $\alpha = O_\varepsilon(\sqrt{\log n \log(\max(1/\rho, 1/\tau))})$, $\beta = 1 + \varepsilon$ and $\gamma = O(1)$.*

Proof. Let S^* be an optimal solution to $\langle G, y, w, \tau, \rho \rangle$, note that $|S^*| \leq \rho n$, $y(S^*) \geq \tau \cdot y(V)$ and $\delta(S^*) = \mathsf{OPT}_{\langle G, y, w, \tau, \rho \rangle}$ the optimal value of this instance. Define two measures on V as follows. For any $S \subseteq V$, set $\mu(S) := |S|/n$ and $\eta(S) := y(S)/y(V)$.

The algorithm guesses $H \geq \tau$ such that $H \leq \eta(S^*) \leq 2H$ (see Algorithm I above for an argument why we can guess H). Then it invokes the algorithm from part II on G with measures μ and η as defined above, and parameters ρ, H. The obtained solution S satisfies $|S| = \mu(S) \cdot n \leq (1 + \varepsilon)\rho\, n$ and $y(S) = \eta(S) \cdot y(V) \geq \Omega_\varepsilon(1)\, H \cdot y(V) \geq \Omega_\varepsilon(1)\, \tau \cdot y(V)$, since $H \geq \tau$. Furthermore, $\delta(S) \leq \alpha \cdot \delta(S^*) \cdot \eta(S)/\eta(S^*) \leq \alpha \cdot \delta(S^*) \cdot \Theta_\varepsilon(1)$, where $\alpha = O_\varepsilon(\sqrt{\log n \log(\max(1/\rho, 1/\tau))})$. $\qquad\square$

3 Min-max Balanced Partitioning

In this section, we present our algorithm for Min–Max k–Partitioning, assuming a subroutine that approximates Weighted ρ-Unbalanced Cut (which is essentially a rephrasing of Weighted Small-Set Expansion). Our algorithm for Min–Max k–Partitioning follows by a straightforward composition of Theorem 3.1 and Theorem 3.3 below. Plugging in for (α, β, γ) the values obtained in Section 2 would complete the proof of Theorem 1.1.

3.1 Uniform Coverings

We first consider a covering relaxation of Min–Max k–Partitioning and solve it using multiplicative updates. This covering relaxation can alternatively be viewed as a fractional solution to a configuration LP of exponential size, as discussed further below.

Let $\mathcal{C} = \{S \subseteq V : |S| \leq n/k\}$ denote all the vertex-sets that are feasible for a single part. Note that a feasible solution in Min–Max k–Partitioning corresponds to a partition of V into k parts, where each part belongs to $\mathcal{C}$. Algorithm 1, described below, *uniformly covers* V using sets in $\mathcal{C}$ (actually a slightly larger family than $\mathcal{C}$). It is important to note that its output $\mathcal{S}$ is a multiset.

Algorithm 1: Covering Procedure for Min–Max k–Partitioning:

Set $t = 1$, and $y^1(v) = 1$ for all $v \in V$

while $\sum_{v \in V} y^t(v) > 1/n$ **do**

     `// Solve the following using algorithm from Corollary 2.7.`

     Let $S^t \subseteq V$ be the solution for Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k} \rangle$.

     Set $\mathcal{S} = \mathcal{S} \cup \{S^t\}$.

     `// Update the weights of the covered vertices.`

     for every $v \in V$ **do**

         Set $y^{t+1}(v) = \frac{1}{2} \cdot y^t(v)$ if $v \in S^t$, and $y^{t+1}(v) = y^t(v)$ otherwise.

     Set $t = t + 1$.

return $\mathcal{S}$

Theorem 3.1. *Running Algorithm 1 on an instance of Min–Max k–Partitioning outputs $\mathcal{S}$ that satisfies (here OPT denotes the optimal value of the instance):*

1. *For all $S \in \mathcal{S}$ we have $\delta(S) \leq \alpha \cdot$ OPT and $|S| \leq \beta \cdot n/k$.*

2. *For all $v \in V$ we have $|\{S \in \mathcal{S} : S \ni v\}|/|\mathcal{S}| \geq 1/(5\gamma k)$.*

Proof. For an iteration t, let us denote $Y^t := \sum_{v \in V} y^t(v)$. The first assertion of the theorem is immediate from the following claim.

Claim 3.2. *Every iteration t of Algorithm 1 satisfies $\delta(S^t) \leq \alpha \cdot$ OPT and $|S^t| \leq \beta \cdot n/k$.*

Proof. It suffices to show that the optimal value of the Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k} \rangle$ is at most OPT. To see this, consider the optimal solution $\{S_i^*\}_{i=1}^k$ of the original Min–Max k–Partitioning instance. We have $|S_i^*| \leq n/k$ and $w(\delta(S_i^*)) \leq$ OPT for all $i \in [k]$. Since $\{S_i^*\}_{i=1}^k$ partitions V, there is some $j \in [k]$ with $y^t(S_j^*) \geq Y^t/k$. It now follows that S_j^* is a feasible solution to the Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k} \rangle$, with objective value at most OPT, which proves the claim. $\square$

We proceed to prove the second assertion of Theorem 3.1. Let ℓ denote the number of iterations of the while loop, for the given Min–Max k–Partitioning instance. For any $v \in V$, let N_v denote the number of iterations t with $S^t \ni v$. Then, by the y-updates we have $y^{\ell+1}(v) = 1/2^{N_v}$. Moreover, the termination condition implies that $y^{\ell+1}(v) \leq 1/n$ (since $Y^{\ell+1} \leq 1/n$). Thus we obtain $N_v \geq \log_2 n$ for all $v \in V$. From the approximation guarantee of the Weighted ρ-Unbalanced Cut algorithm, it follows that $y^t(S^t) \geq \frac{1}{\gamma k} \cdot Y^t$ in every iteration t. Thus $Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \leq \left(1 - \frac{1}{2\gamma k}\right) \cdot Y^t$. This implies that $Y^\ell \leq \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} \cdot Y^1 = \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} \cdot n$. However $Y^\ell > 1/n$ since the algorithm performs ℓ iterations. Thus, $\ell \leq 1 + 4\gamma k \cdot \ln n \leq 5\gamma k \cdot \log_2 n$. This proves $|\{S \in \mathcal{S} : S \ni v\}|/|\mathcal{S}| = N_v/\ell \geq (5\gamma)^{-1} k^{-1}$. $\square$

Alternative view: A configuration LP. We now describe an alternate approach to finding a cover $\mathcal{S}$. Given a bound λ on the cost of any single cut, define the set of feasible cuts as follows:

$$\mathcal{F}_\lambda = \left\{ S \subseteq V \ : \ |S| \leq \frac{n}{k}, \ \delta(S) \leq \lambda \right\}.$$

We define a configuration LP for Min–Max k–Partitioning as follows. There is a variable x_S for each $S \in \mathcal{F}_\lambda$ indicating whether/not cut S is chosen.

$$
\begin{array}{rll}
\mathcal{P}(\lambda) = & \min & \sum_{S \in \mathcal{F}_\lambda} x_S \\
& \text{s.t.} & \sum_{S \in \mathcal{F}_\lambda : v \in S} x_S \geq 1 \quad \forall v \in V \\
& & x_S \geq 0 \quad\quad\quad\quad \forall S \in \mathcal{F}_\lambda
\end{array}
\tag{8}
$$

The goal is determine the smallest $\lambda > 0$ such that $\mathcal{P}(\lambda) \leq k$. One can approximately solve this using the dual formulation:

$$
\begin{array}{rll}
\mathcal{D}(\lambda) = & \max & \sum_{v \in V} y_v \\
& \text{s.t.} & \sum_{v \in S} y_v \leq 1 \quad \forall S \in \mathcal{F}_\lambda \\
& & y_v \geq 0 \quad\quad \forall v \in V
\end{array}
\tag{9}
$$

The dual separation oracle can be solved using Weighted Small-Set Expansion; so we can apply the Ellipsoid algorithm. Since we only have a multi-criteria approximation for Weighted Small-Set Expansion (see Section 2), the details for approximating the configuration LP are rather technical.

3.2 Aggregation

The aggregation process, which might be of independent interest, transforms a cover of G into a partition. Intuitively, we first let the sets randomly compete with each other over the vertices so as to form a partition; then, to make sure no set has large cost, we repeatedly fix the partition locally, and use a potential function to track progress.

Theorem 3.3. *Algorithm 2 is a randomized polynomial-time algorithm that when given a graph $G = (V, E)$, an $\varepsilon \in (0, 1)$, and a cover $\mathcal{S}$ of V that satisfies: (i) every vertex in V is covered by at least c/k fraction of sets $S \in \mathcal{S}$, for $c \in (0, 1]$; and (ii) all $S \in \mathcal{S}$ satisfy $|S| \leq 2n/k$ and $\delta(S) \leq B$; the algorithm outputs a partition $\mathcal{P}$ of V into at most k sets such that for all $P \in \mathcal{P}$ we have $|P| \leq 2(1 + \varepsilon)n/k$ and $\mathbb{E}[\max \delta(P) : P \in \mathcal{P}] \leq 8B/(c\varepsilon)$.*

Analysis. 1. Observe that after step 1 the collection of sets $\{P_i\}$ is a partition of V and $P_i \subset S_i$ for every i. Particularly, $|P_i| \leq |S_i| \leq 2n/k$. Note, however, that the bound $\delta(P_i) \leq B$ may be violated for some i. We now prove that $\mathbb{E}\left[\sum_i \delta(P_i)\right] \leq 2kB/c$. Fix an $i \leq |\mathcal{S}|$ and estimate the expected weight of edges $E(P_i, \cup_{j>i} P_j)$ given that $S_i = S$. If an edge (u, v) belongs to $E(P_i, \cup_{j>i} P_j)$ then $(u, v) \in E(S_i, V \setminus S_i) = E(S, V \setminus S)$ and both $u, v \notin \cup_{j<i} S_j$. For any edge $(u, v) \in \delta(S)$ (with $u \in S$,

---

Algorithm 2: Aggregation Procedure for Min–Max k–Partitioning:

---

1 Sampling

    Sort sets in $\mathcal{S}$ in a random order: $S_1, S_2, \ldots, S_{|\mathcal{S}|}$. Let $P_i = S_i \setminus \cup_{j<i} S_j$.

2 Replacing Expanding Sets with Sets from $\mathcal{S}$

    while *there is a set P_i such that $\delta(P_i) > 2B$* **do**

    ⌊   Set $P_i = S_i$, and for all $j \neq i$, set $P_j = P_j \setminus S_i$.

3 Aggregating

    Let $B' = \max\{\frac{1}{k} \sum_i \delta(P), 2B\}$.

    while *there are $P_i \neq \varnothing$, $P_j \neq \varnothing$ $(i \neq j)$ such that $|P_i| + |P_j| \leq 2(1+\varepsilon)n/k$ and*
    $\delta(P_i) + \delta(P_j) \leq 2B'\varepsilon^{-1}$ **do**

    ⌊   Set $P_i = P_i \cup P_j$ and set $P_j = \varnothing$.

4 return all non-empty sets P_i.

$v \notin S$), $\Pr((u,v) \in E(P_i, \cup_{j>i} P_j) \mid S_i = S) \leq \Pr(v \notin \cup_{j<i} S_j \mid S_i = S) \leq (1 - c/k)^{i-1}$, since v is covered by at least c/k fraction of sets in $\mathcal{S}$ and is not covered by $S_i = S$. Hence,

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j)) \mid S_i = S] \leq (1 - c/k)^{i-1}\delta(S) \leq (1 - c/k)^{i-1}B,$$

and $\mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \leq (1 - c/k)^{i-1}B$. Therefore, the total expected weight of edges crossing the boundary of P_i's is at most $\sum_{i=0}^{\infty}(1 - c/k)^i B = kB/c$, and $\mathbb{E}\left[\sum_i \delta(P_i)\right] \leq 2kB/c$.

2. After each iteration of step 2, the following invariant holds: the collection of sets $\{P_i\}$ is a partition of V and $P_i \subset S_i$ for all i. Particularly, $|P_i| \leq |S_i| \leq 2n/k$. The key observation is that at every iteration of the "while" loop, the sum $\sum_j \delta(P_j)$ decreases by at least $2B$. This is due to the following uncrossing argument:

$$
\begin{aligned}
\delta(S_i) + \sum_{j \neq i} \delta(P_j \setminus S_i) &\leq \delta(S_i) + \sum_{j \neq i}\Big(\delta(P_j) + w(E(P_j \setminus S_i, S_i)) - w(E(S_i \setminus P_j, P_j))\Big) \\
&\leq \delta(S_i) + \Big(\sum_{j \neq i}\delta(P_j)\Big) + \underbrace{w(E(V \setminus S_i, S_i))}_{\delta(S_i)} - \underbrace{w(E(P_i, V \setminus P_i))}_{\delta(P_i)} \\
&= \Big(\sum_j \delta(P_j)\Big) + 2\delta(S_i) - 2\delta(P_i) \leq \Big(\sum_j \delta(P_j)\Big) - 2B.
\end{aligned}
$$

we used that $P_i \subset S_i$, all P_j are disjoint, $\cup_{j \neq i}(P_j \setminus S_i) \subset V \setminus S_i$, $P_i \subset S_i \setminus P_j$, $\cup_{j \neq i} P_j = V \setminus P_i$. Hence, the number of iterations of the loop in step 2 is always polynomially bounded and after the last iteration $\mathbb{E}\left[\sum_i \delta(P_i)\right] \leq 2kB/c$ (the expectation is over random choices at step 1; the step 2 does not use random bits). Hence, $\mathbb{E}[B'] \leq 4B/c$.

3. The following analysis holds conditional on any value of B'. After each iteration of step 3, the following invariant holds: the collection of sets $\{P_i\}$ is a partition of V. Moreover, $|P_i| \leq 2(1+\varepsilon)n/k$ and $\delta(P_i) \leq 2B'\varepsilon^{-1}$ (note: after step 2, $\delta(P_i) \leq 2B \leq B'$ for each i).

When the loop terminates, we obtain a partition of V into sets P_i satisfying $|P_i| \leq 2(1+\varepsilon)n/k$, $\sum_i |P_i| = n$, $\delta(P_i) \leq 2B'\varepsilon^{-1}$, $\sum_i \delta(P_i) \leq kB'$, such that no two sets can be merged without violating above constraints. Hence by Lemma 3.4 below (with $a_i = |P_i|$ and $b_i = \delta(P_i)$), the number of non-empty sets is at most $2\frac{n}{2(1+\varepsilon)n/k} + \frac{kB'}{2B'\varepsilon^{-1}} = (1+\varepsilon)^{-1}k + (\varepsilon/2)k \leq k$. □

Lemma 3.4 (Greedy Aggregation). *Let $a_1, \ldots, a_t$ and $b_1, \ldots b_t$ be two sequences of nonnegative numbers satisfying the following constraints $a_i < A$, $b_i < B$, $\sum_{i=1}^{t} a_i \leq S$ and $\sum_{i=1}^{t} b_i \leq T$ (for some positive real numbers A, B, S, and T). Moreover, assume that for every i and j $(i \neq j)$ either $a_i + a_j > A$ or $b_i + b_j > B$. Then, $t < S/A + T/B + \max(S/A, T/B, 1)$.*

Proof. By rescaling we assume that $A = 1$ and $B = 1$. Moreover, we may assume that $\sum_{i=1}^{t} a_i < S$ and $\sum_{i=1}^{t} b_i < T$ by slightly decreasing values of all a_i and b_i so that all inequalities still hold.

We write two linear programs. The first LP (LP_I) has variables x_i and constraints $x_i + x_j \geq 1$ for all i, j such that $a_i + a_j \geq 1$. The second LP (LP_{II}) has variables y_i and constraints $y_i + y_j \geq 1$ all i, j such that $b_i + b_j \geq 1$. The LP objectives are to minimize $\sum_i x_i$ and to minimize $\sum_i y_i$. Note, that $\{a_i\}$ is a feasible point for LP_I and $\{b_i\}$ is a feasible point for LP_{II}. Thus, the optimum values of LP_I and LP_{II} are strictly less than S and T respectively.

Observe that both LPs are half-integral. Consider optimal solutions x_i^*, y_j^* where $x_i^*, y_j^* \in \{0, 1/2, 1\}$. Note that for every i, j either $x_i^* + x_j^* \geq 1$ or $y_i^* + y_j^* \geq 1$. Consider several cases. If for all i, $x_i^* + y_i^* \geq 1$, then $t < S + T$, since $\sum_{i=1}^{t}(x_i^* + y_i^*) < S + T$. If for some j, $x_j^* + y_j^* = 0$ (and hence $x_j^* = y_j^* = 0$), then $x_i^* + y_i^* \geq 1$ for $i \neq j$ and, thus, $t < S + T + 1$. Finally, assume that for some j, $x_j^* + y_j^* = 1/2$, and w.l.o.g. $x_j^* = 1/2$ and $y_j^* = 0$. The number of i's with $x_i^* \neq 0$ is (strictly) bounded by $2S$. For the remaining i's, $x_i^* = 0$ and hence $y_i^* = 1$ (because $y_i^* = y_i^* + y_j^* \geq 1$), and thus the number of such i's is (strictly) bounded by T. □

4 Further Extensions

Both Theorems 1.1 and 1.4 follow from a more general result for a problem that we call Min–Max Cut, defined as follows. The input is an undirected graph $G = (V, E)$, nonnegative edge-weights w, a collection of disjoint terminal sets $T_1, T_2, \ldots, T_k \subset V$ (possibly empty), and parameters $\rho \in [1/k, 1]$ and $C, D > 0$. The goal is to find a partition $S_1, \ldots, S_k$ of V such that:

1. For all i, $T_i \subseteq S_i$;
2. For all i, $|S_i| \leq \rho n$;
3. For all i, $\delta(S_i) \leq C$; and
4. $\sum_i \delta(S_i) \leq D$.

This problem models the aforementioned cloud computing scenario, where in addition, certain processes are preassigned to machines (each set T_i maps to machine $i \in [k]$). The goal is to assign the processes V to machines $[k]$ while respecting the preassignment and machine load constraints, and minimizing both bandwidth per machine and total volume of communication.

Theorem 4.1. *There is a randomized polynomial time algorithm that given any feasible instance of the Min–Max Cut problem with parameters k, ρ, C, D and any $\varepsilon > 0$, finds a partition $Q_1, \ldots, Q_k$ with the following properties: (i) For all i, $T_i \subseteq Q_i$; (ii) For all i, $|Q_i| \leq (2 + \varepsilon)\rho n$; (iii) $\mathbb{E}\left[\max_{i=1}^{k} \delta(Q_i)\right] \leq O_\varepsilon(\sqrt{\log n \log k})C$; and (iv) $\mathbb{E}\left[\sum_i \delta(Q_i)\right] \leq O_\varepsilon(\sqrt{\log n \log k})D$.*

It is clear that in fact Theorem 4.1 generalizes both Theorems 1.1 and 1.4. Let us now describe modifications to the Min–Max k–Partitioning algorithm used to obtain Theorem 4.1.

Uniform Coverings. First, by the introduction of vertex weights, we can shrink each preassigned set T_i to a single terminal t_i (for $i \in [k]$). Then, feasible vertex-sets $\mathcal{C}$ in the covering procedure (Section 3.1) consist of those $S \subseteq V$ where $\mathrm{weight}(S) \leq \rho n$ (balance constraint) and $|S \cap \{t_i\}_{i=1}^k| \leq 1$ (preassignment constraint). The subproblem Weighted ρ-Unbalanced Cut also has the additional $|S \cap \{t_i\}_{i=1}^k| \leq 1$ constraint; this can be handled in the algorithm from Section 2 by guessing which terminal belongs to S (see Remark 2.3). Using Corollary 2.7 we assume an (α, β, γ) approximation algorithm for this (modified) Weighted ρ-Unbalanced Cut problem; where for any $\varepsilon > 0$, $\alpha = O_\varepsilon(\sqrt{\log n \, \log(\max\{1/\rho, 1/\tau\})})$, $\beta = 1 + \varepsilon$ and $\gamma = O_\varepsilon(1)$.

Algorithm 3 below gives the procedure to obtain a uniform covering $\mathcal{S}$ bounding total edge-cost in addition to the conditions in Theorem 3.1.

Algorithm 3: Covering Procedure for Min–Max Cut:

set $t \leftarrow 1$, $y^1(v) \leftarrow 1$ for all $v \in V$, and $Y^1 \leftarrow \sum_{v \in V} y^1(v)$.
while $Y^t > \frac{1}{n}$ **do**
    for $i = 0, \ldots, \log_2 k + 1$ **do**
        solve the Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{2^i}, \rho \rangle$ using the algorithm
        from Corollary 2.7, to obtain $S^t(i) \subseteq V$.
        If $\big(\delta(S^t(i)) \leq \alpha \cdot \min\{C, 4D/2^i\} \big)$ then $S^t \leftarrow S^t(i)$ and quit for loop.
    set $\mathcal{S} \leftarrow \mathcal{S} \cup \{S^t\}$.
    for $v \in V$ **do**
        set $y^{t+1}(v) \leftarrow \frac{1}{2} \cdot y^t(v)$ if $v \in S^t$, and $y^{t+1}(v) \leftarrow y^t(v)$ otherwise.
    set $Y^{t+1} \leftarrow \sum_{v \in V} y^{t+1}(v)$.
    set $t \leftarrow t + 1$.
output $\mathcal{S}$.

Theorem 4.2. *For any instance of Min–Max Cut, output $\mathcal{S}$ of Algorithm 3 satisfies:*

1. $\delta(S) \leq \alpha \cdot C$ and $|S| \leq \beta \cdot \frac{n}{k}$ for all $S \in \mathcal{S}$.

2. $|\{S \in \mathcal{S} : S \ni v\}| \geq \log_2 n$ for all $v \in V$.

3. $|\mathcal{S}| \leq 5\gamma\, k \cdot \log_2 n$.

4. $\sum_{S \in \mathcal{S}} \delta(S) \leq 17\alpha\, \gamma\, \log_2 n \cdot D$.

Above, for any $\varepsilon > 0$, $\alpha = O_\varepsilon(\sqrt{\log n \, \log k})$, $\beta = 1 + \varepsilon$ and $\gamma = O_\varepsilon(1)$.

Proof. We start with the following key claim.

Claim 4.3. *In any iteration t of the above algorithm, there exists an $i \in \{0, 1, \ldots, \log_2 k + 1\}$ such that $\delta(S^t(i)) \leq \alpha \cdot \min\{C, 4D/2^i\}$, $|S^t(i)| \leq \beta \cdot \frac{n}{k}$, and $y^t(S^t(i)) \geq \frac{Y^t}{\gamma\, 2^i}$.*

Proof. Consider the optimal solution $\{S_j^*\}_{j=1}^k$ of the original Min–Max Cut instance. For all $j \in [k]$ we have that $|S_j^*| \leq \rho n$, $\delta(S_j^*) \leq C$ and S_j^* contains at most one terminal. Moreover, $\sum_{j=1}^k \delta(S_j^*) \leq D$. Since $\{S_j^*\}_{j=1}^k$ partitions V, we also have $\sum_{j=1}^k y^t(S_j^*) = Y^t$. Let $L \subseteq [k]$ denote the indices j having $\delta(S_j^*) \leq \frac{2D}{Y^t} \cdot y^t(S_j^*)$.

20

We claim that $\sum_{j \in L} y^t(S_j^*) \geq Y^t/2$. This is because:

$$\sum_{j \notin L} y^t(S_j^*) \leq \frac{Y^t}{2D} \cdot \sum_{j \notin L} \delta(S_j^*) \leq \frac{Y^t}{2D} \cdot \sum_{j=1}^{k} \delta(S_j^*) \leq Y^t/2.$$

Since $|L| \leq k$, there is some $q \in L$ with $y^t(S_q^*) \geq \frac{Y^t}{2k}$. Let $i \in \{1, \ldots, \log_2 k + 1\}$ be the value such that $y^t(S_q^*)/Y^t \in [\frac{1}{2^i}, \frac{1}{2^{i-1}}]$; note that such an i exists because $y^t(S_q^*)/Y^t \in [\frac{1}{2k}, 1]$. For this i, consider the Weighted ρ-Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{2^i}, \rho \rangle$. Observe that S_q^* is a feasible solution here since $y^t(S_q^*) \geq Y^t/2^i$, $|S_q^*| \leq \rho n$ and S_q^* contains at most one terminal. Hence the optimal value of this instance is at most:

$$\delta(S_q^*) \leq \min \left\{ C, \frac{2D}{Y^t} \cdot y^t(S_q^*) \right\} \leq \min \left\{ C, \frac{4D}{2^i} \right\}$$

The first inequality uses the definition of L and that $\delta(S_q^*) \leq C$, and the second inequality is by choice of i. It now follows from Corollary 2.7 that solution $S^t(i)$ satisfies the claimed properties. We note that $\alpha = O_\varepsilon(\sqrt{\log n \log k})$ because each instance of Weighted ρ-Unbalanced Cut has parameters $\tau = \frac{1}{2^i} \geq \frac{1}{k}$ and $\rho \geq \frac{1}{k}$. $\qquad \square$

For each iteration t, let $i_t \in \{1, \ldots, \log_2 k + 1\}$ be the index such that $S^t = S^t(i_t)$. Claim 4.3 implies that such an index always exists, and so the algorithm is indeed well-defined. **Condition 1** of Theorem 4.2 also follows directly. Let ℓ denote the number of iterations of the while loop, for the given Min–Max Cut instance. For any $v \in V$, let N_v denote the number of iterations t with $S^t \ni v$. Then, by the y-updates we have $y^{\ell+1}(v) = 1/2^{N_v}$. Moreover, the termination condition implies that $y^{\ell+1}(v) \leq \frac{1}{n}$ (since $Y^{\ell+1} \leq \frac{1}{n}$). Thus we obtain $N_v \geq \log_2 n$ for all $v \in V$, proving **condition 2** of Theorem 4.2.

Claim 4.3 implies that for each iteration t, we have $y^t(S^t) \geq \frac{Y^t}{\gamma 2^{i_t}}$ and $\delta(S^t) \leq 4\alpha D/2^{i_t}$. Since $i_t \leq \log_2 k + 1$, we obtain:

$$y^t(S^t) \geq \max \left\{ \frac{Y^t}{2\gamma k}, \frac{Y^t}{4\alpha\gamma D} \cdot \delta(S^t) \right\}.$$

1. Using $y^t(S^t) \geq \frac{Y^t}{2\gamma k}$ in each iteration, $Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \leq \left(1 - \frac{1}{4\gamma k}\right) \cdot Y^t$. This implies that $Y^\ell \leq \left(1 - \frac{1}{4\gamma k}\right)^{\ell-1} \cdot Y^1 = \left(1 - \frac{1}{4\gamma k}\right)^{\ell-1} \cdot n$. However $Y^\ell > \frac{1}{n}$ since the algorithm performs ℓ iterations. Thus, $\ell \leq 1 + 8\gamma k \cdot \ln n \leq 9\gamma k \cdot \log_2 n$. This proves **condition 3** in Theorem 4.2.

2. Using $y^t(S^t) \geq \frac{Y^t}{4\alpha\gamma D} \cdot \delta(S^t)$ in each iteration, $Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \leq \left(1 - \frac{\delta(S^t)}{8\alpha\gamma D}\right) \cdot Y^t$. So,

$$\frac{1}{n} < Y^\ell \leq \Pi_{t=1}^{\ell-1} \left(1 - \frac{\delta(S^t)}{8\alpha\gamma D}\right) \cdot Y^1 \leq \exp\left(-\frac{\sum_{t=1}^{\ell-1} \delta(S^t)}{8\alpha\gamma D}\right) \cdot n$$

This implies $\sum_{t=1}^{\ell-1} \delta(S^t) \leq (16\alpha\gamma \ln n) \cdot D$. Adding in $\delta(S^\ell) \leq \alpha \cdot C \leq \alpha D$, we obtain **condition 4** of Theorem 4.2.

This completes the proof of Theorem 4.2. $\qquad \square$

Aggregation This step remains essentially the same as in Section 3.2, namely Algorithm 3 (with parameter $B := \alpha \cdot C$). The only difference is that in Step 3 we do not merge parts containing terminals. We first show that this yields a slightly weaker version of Theorem 4.1: in condition (ii) we obtain a bound of $(3 + \varepsilon)\rho n$ on the cardinality of each part. (Later we show how to achieve the cardinality bound of $(2 + \varepsilon)\rho n$ as claimed in Theorem 4.1.)

Note that each of the final sets $\{P_i\}$ is a subset of some set in $\mathcal{S}$, and hence contains at most one terminal. It also follows that the final sets $\{P_i\}$ are at most $2k$ in number: at most k of them contain no terminals (just as in Theorem 3.3), and at most k contain a terminal (since there are at most k terminals). Each of these sets $\{P_i\}$ has size at most $(2 + \varepsilon)\rho n$ and cut value at most $8B/(c\varepsilon)$, by the analysis in Theorem 4.1. Moreover, if a set P_i contains a terminal then $|P_i| \leq \beta \cdot \rho n = (1 + \varepsilon)\rho n$ (since it does not participate in any merge). Finally in order to reduce the number of parts to k, we merge arbitrarily each part containing a terminal with one non-terminal part; and output this as the final solution. It is clear that each part has at most one terminal, has size $\leq (3 + \varepsilon)\rho n$, and cut value at most $O_\varepsilon(\sqrt{\log n \log k}) \cdot C$. The bound on total cost (condition (iv) in Theorem 4.1) is by the following claim. This proves a weaker version of Theorem 4.1, with size bound $(3 + \varepsilon)\rho n$.

Claim 4.4. *Algorithm 3 applied on collection $\mathcal{S}$ from Theorem 4.2 outputs partition $\{P_i\}_{i=1}^k$ satisfying $\mathbb{E}\left[\sum_{i=1}^k w(\delta(P_i))\right] = O_\varepsilon(\sqrt{\log n \log k}) \, D$.*

Proof. We will show that the random partition $\{P_i\}$ at the end of Step 1 in Algorithm 3 satisfies $\mathbb{E}\left[\sum_i w(\delta(P_i))\right] \leq O_\varepsilon(\sqrt{\log n \log k}) \, D$. This would suffice since $\sum_i w(\delta(P_i))$ does not increase in Steps 2 and 3. For notational convenience, we assume (by adding empty sets) that $|\mathcal{S}| = 5\gamma k \cdot \log_2 n$ in Theorem 4.2; note that this does not affect any of the other conditions.

To bound the cost of the partition $\{P_i\}$ in Step 1, consider any index $i \leq |\mathcal{S}|$. From the proof of Theorem 3.3, we have:

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j)) \mid S_i = S] \leq (1 - c/k)^{i-1} \delta(S),$$

where $c = \frac{1}{5\gamma}$ is such that each vertex lies in at least c/k fraction of sets $\mathcal{S}$. Deconditioning, $\mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \leq (1 - c/k)^{i-1} \cdot \mathbb{E}[\delta(S_i)] = (1 - c/k)^{i-1} \cdot \left(\sum_{S \in \mathcal{S}} \delta(S)/|\mathcal{S}|\right)$, where we used that S_i is a uniformly random set from $\mathcal{S}$. So the total edge-cost,

$$\mathbb{E}\left[\sum_i \delta(P_i)\right] = 2 \cdot \sum_i \mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \leq \left(\sum_{i \geq 0}(1 - c/k)^i\right) \cdot \left(\sum_{S \in \mathcal{S}} \delta(S)/|\mathcal{S}|\right) = \frac{k}{c} \cdot \sum_{S \in \mathcal{S}} \delta(S)/|\mathcal{S}|.$$

Using $\sum_{S \in \mathcal{S}} \delta(S) \leq 17\alpha\gamma \log_2 n \cdot D$ and $|\mathcal{S}| = 5\gamma k \cdot \log_2 n$ from Theorem 4.2, $\mathbb{E}\left[\sum_i \delta(P_i)\right] \leq \frac{17\alpha}{5c} D = O_\varepsilon(\sqrt{\log n \log k}) \, D$ since $\alpha = O_\varepsilon(\sqrt{\log n \log k})$ and $1/c = O_\varepsilon(1)$. □

Obtaining size bound of $(2 + \varepsilon)\rho n$. We now describe a modified aggregating step (in place of Step 3 in Algorithm 3) that yields Theorem 4.1. Given the uniform cover $\mathcal{S}$ from Algorithm 3, run Steps 1 and 2 of Algorithm 3 (use $B = \alpha C$) to obtain parts $P_1, \ldots, P_{|\mathcal{S}|}$. Then:

1. Set $B' := \max\left\{\frac{1}{k} \sum_i \delta(P_i), \, 2B\right\}$.

2. While there are $P_i, P_j \neq \varnothing$ ($i \neq j$) such that $|P_i| + |P_j| \leq (1 + \varepsilon)\rho n$, $\delta(P_i) + \delta(P_j) \leq 2B'$ and $P_i \cup P_j$ does not contain a terminal: replace $P_i \leftarrow P_i \cup P_j$ and $P_j \leftarrow \varnothing$.

3. Sort the resulting non-empty sets $P_1, \ldots, P_t$ in non-increasing order of size.

4. Form $\lceil t/k \rceil$ groups where the jth group consists of parts indexed between $(j-1)k+1$ and jk.

5. For each $i \in [k]$ define Q_i as the union of one part from each group such that it contains terminal i but no other terminal. Additionally, ensure that each part is assigned to one of $\{Q_i\}_{i=1}^k$.

We first show that the number of parts after Step 2 above $t \leq 4k$. Note that each part contains at most one terminal, and the number of parts containing a terminal is at most k. For the non-terminal parts, using Lemma 3.4 (with $a_i = |P_i|$, $b_i = \delta(P_i)$, $a = (1+\varepsilon)\rho n$, $b = 2B'$, $S = n$ and $T = kB'$) we obtain a bound of $5k/2$, which implies $t \leq 4k$.

Next observe that sets $\{Q_i\}_{i=1}^k$ in Step 5 above are well-defined, and can be found using a simple greedy rule: this is because each group contains at most k parts. This gives Theorem 4.1 (i). Since $t \leq 4k$ there are at most 4 groups and hence $\max_{i=1}^k \delta(Q_i) \leq 4 \cdot \max_{i=1}^t \delta(P_i) \leq 8B'$. This proves Theorem 4.1 (iii). Also, the proof of Claim 4.4 implies Theorem 4.1 (iv); this is due to the fact that in the final partition $\sum_{i=1}^k \delta(Q_i) \leq \sum_{\ell=1}^{|\mathcal{S}|} \delta(P_\ell)$.

We now show Theorem 4.1 (ii), i.e. $\max_{i=1}^k |Q_i| \leq (2+\varepsilon)\rho n$. Consider any $i \in [k]$ and let P_i' denote the part assigned to Q_i from the first group. By the non-increasing size ordering $P_1, \ldots, P_t$ and the round-robin assignment (Step 5) into Q_is, we obtain that $|Q_i| - |P_i'| \leq |Q_j|$ for all $j \in [k]$. Taking an average, $|Q_i| - |P_i'| \leq \frac{1}{k} \sum_{j=1}^k |Q_j| = \frac{n}{k}$. Finally, since each part $\{P_\ell\}_{\ell=1}^t$ has size at most $(1+\epsilon)\rho n$ we have $|Q_i| \leq (2+\epsilon)\rho n$. This completes the proof of Theorem 4.1.

5 Hardness of Min-Max-Multiway-Cut

In this section we prove Theorem 1.5, which shows that obtaining a $k^{1-\varepsilon}$-approximation algorithm for Min-Max-Multiway-Cut is hard, if not unlikely. This suggests that some dependence on n might be necessary (unless we are satisfied with approximation that is linear in k, which is trivial), which is in contrast to several cut problems (multiway-cut, multicut, requirement-cut etc) with sum-objective where $poly(\log k)$ approximation guarantees are known when k is the size of the terminal set [Moi09, LM10, EGK$^+$10, CLLM10, MM10]. Throughout this section, we assume k is constant (independent of n), which simplifies the statements; strictly speaking, our reductions relate solving one problem with parameter $k(n)$ to solving another problem with parameter $k(n')$.

We will refer to the min-sum version of Min–Max k–Partitioning, called Min–Sum k–Partitioning, in which the input is an edge-weighted graph $G = (V, E)$ and a parameter k, and the goal is to partition the vertices into k equal-sized parts while minimizing the total edge-weight of all edges cut. An algorithm for Min–Sum k–Partitioning is an (α, β) bicriteria approximation if for every instance, it partitions the vertices into k pieces, each of size at most $\beta|V|/k$, and the total edge-weight of all edges cut is at most α times the least possible among all partitions into k equal-size sets.

The basic idea in proving Theorem 1.5 as follows. although there is no vertex-balance requirement in Min-Max-Multiway-Cut, an edge-balance is implicit in the objective. By introducing a complete bipartite graph (having suitable edge weight) between the terminals and the rest of the graph, this edge-balance can be used to enforce the vertex-balance required in Min–Sum k–Partitioning. After this first step, which obtains a bicriteria approximation for Min–Sum k–Partitioning, we do a second step which improves the size-violation in the algorithm for Min–Sum k–Partitioning.

These two steps are formalized in the two foregoing Lemmas 5.1 and 5.3. Putting them together immediately proves 1.5.

Lemma 5.1. *If there is a ρ-approximation algorithm for Min-Max-Multiway-Cut then there is a $(5\rho\,k,\,10\rho)$-bicriteria approximation algorithm for Min–Sum k–Partitioning.*

Proof. Let $\mathcal{A}$ denote the ρ-approximation algorithm for Min-Max-Multiway-Cut. Consider any instance $\mathcal{I}$ of Min–Sum k–Partitioning: graph $G = (V, E)$, edge costs $c : E \to \mathbb{R}_+$ and parameter k. For every $B \geq 0$, consider an instance $\mathcal{J}(B)$ of Min-Max-Multiway-Cut on graph G_B defined as:
- The vertex set is $V \bigcup \{t_i\}_{i=1}^{k}$ where $\{t_i\}_{i=1}^{k}$ are the terminals.
- The edges are $E \bigcup \{(t_i, u) : i \in [k], u \in V\}$.
- Extend cost function c by setting $c(t_i, u) = \frac{B}{n}$ for all $i \in [k]$, $u \in V$.

Note that every solution to $\mathcal{J}(B)$ corresponds to a k–partition in G (though possibly unbalanced). We say that a solution to $\mathcal{J}(B)$ is β-balanced if each piece in the partition has size at most $\beta \cdot \frac{n}{k}$.

The algorithm for Min–Sum k–Partitioning on $\mathcal{I}$ runs algorithm $\mathcal{A}$ on all the Min-Max-Multiway-Cut instances $\left\{ \mathcal{J}(2^i) : 0 \leq i \leq \log_2\left(\sum_{e\in E} c_e\right) \right\}$, and returns the cheapest partition that is (10ρ)-balanced. We now show that this results in a $(5\rho\,k,\,10\rho)$ bicriteria approximation ratio.

Note that algorithm $\mathcal{A}$ must be invoked on $\mathcal{J}(B)$ for some value B with $B < \mathsf{OPT}(\mathcal{I}) \leq 2 \cdot B$. We will show that the partition resulting from this call is the desired bicriteria approximation.

Claim 5.2. $\mathsf{OPT}(\mathcal{J}(B)) \leq \mathsf{OPT}(\mathcal{I}) + 2B \leq 5\mathsf{OPT}(\mathcal{I})$.

Proof. Let P^* denote the optimal k–partition to $\mathcal{I}$. Consider the solution to $\mathcal{J}(B)$ obtained by including each terminal into a distinct piece of P^*. The boundary of the piece containing t_i (any $i \in [k]$) in G_B costs at most:

$$\mathsf{OPT}(\mathcal{I}) + n \cdot \frac{B}{n} + \frac{n}{k} \cdot (k-1) \cdot \frac{B}{n} \leq \mathsf{OPT}(\mathcal{I}) + 2B,$$

the term $\mathsf{OPT}(\mathcal{I})$ is due to edges in E, the second term is due to edges at t_i and the third is due to edges at all other $\{t_j : j \neq i\}$. The claim now follows since $B \leq 2\,\mathsf{OPT}(\mathcal{I})$. □

Let P denote $\mathcal{A}$'s solution to $\mathcal{J}(B)$, and $\{P_i\}_{i=1}^{k}$ the partition of V induced by P. From Claim 5.2, P has objective value (for Min-Max-Multiway-Cut) at most $5\rho\,\mathsf{OPT}(\mathcal{I})$. Note that the boundary (in graph G_B) of t_i's piece in P costs *at least* $c(\delta_G(P_i)) + (k-1)\frac{B}{n} \cdot |P_i|$. Thus we obtain:

$$c(\delta_G(P_i)) + (k-1)\frac{B}{n} \cdot |P_i| \leq 5\rho\,\mathsf{OPT}(\mathcal{I}), \qquad \text{for all } i \in [k].$$

It follows that for every $i \in [k]$, we have:

1. $|P_i| \leq \frac{5\rho\,\mathsf{OPT}(\mathcal{I})}{B} \cdot \frac{n}{k-1} \leq 10\rho \cdot \frac{n}{k}$ since $B > \mathsf{OPT}(\mathcal{I})$, and

2. $c(\delta_G(P_i)) \leq 5\rho \cdot \mathsf{OPT}(\mathcal{I})$.

Thus the solution $\{P_i\}_{i=1}^{k}$ to $\mathcal{I}$ is (10ρ)-balanced and costs at most $5\rho\,k \cdot \mathsf{OPT}(\mathcal{I})$. This complete the proof of Lemma 5.1. □

Lemma 5.3. *If there is an $(\alpha, k^{1-\varepsilon})$-bicriteria approximation algorithm for Min–Sum k–Partitioning for some constant $\varepsilon > 0$, then there is also an $(\alpha \cdot \log\log k, \gamma)$-bicriteria approximation algorithm with $\gamma \leq 3^{2/\varepsilon}$.*

24

Proof. Let $\mathcal{A}$ denote the $(\alpha, k^{1-\varepsilon})$-bicriteria approximation algorithm. The idea is to use $\mathcal{A}$ recursively to obtain the claimed $(\alpha \cdot \log \log k, \gamma)$ approximation; details are below. Let G denote the input graph with n vertices, and OPT the optimal balanced k–partition. The algorithm deals with several sub-instances of G, each of which is assigned a *level* from $\{0, 1, \ldots, t\}$ where $t := \Theta(\log \log k)$ is fixed below. Every level i instance will contain $k_i = k^{(1-\varepsilon/2)^i}$ parts and $n_i = \frac{n}{k} \cdot k_i$ vertices. Choose t to be the smallest integer such that $k_t \leq 3^{2/\varepsilon}$. While generating the sub-instances we also add dummy singleton vertices (to keep the instances balanced). For notational simplicity we use the same identifier for a sub-instance and the graph corresponding to it. Note that G is the unique level 0 instance. For each $i \in \{0, 1, \ldots, t-1\}$, every level i instance $\mathcal{I}$ generates k_i level $i+1$ instances as follows:

- Run algorithm $\mathcal{A}$ on $\mathcal{I}$ to obtain a $k_i^{1-\varepsilon}$-balanced partition $\{P_j : 1 \leq j \leq k_i\}$ of $\mathcal{I}$.

- For each $j \in [k_i]$, add $n_{i+1} - |P_j|$ singleton vertices to $\mathcal{I}[P_j]$ to obtain a new level $i+1$ instance.

Note that this process is indeed well-defined since $|P_j| \leq \frac{n_i}{k_i} \cdot k_i^{1-\varepsilon} \leq \frac{n_i}{k_i} \cdot k_i^{1-\varepsilon/2} = \frac{n_i}{k_i} \cdot k_{i+1} = n_{i+1}$. Moreover the total number of level $i+1$ instances is $\Pi_{\ell=0}^i k_\ell = k^{\sum_{\ell=0}^i (1-\varepsilon/2)^\ell} \leq k^{2/\varepsilon}$. Since each instance has size at most n, the total size of all instances is polynomial.

The algorithm finally returns the partition $\mathcal{P}$ corresponding to the set of all level t instances. Note that there are at most $n_t = \frac{n}{k} \cdot k_t \leq 3^{2/\varepsilon} \cdot \frac{n}{k}$ vertices in each level t instance. The algorithm ignores all dummy vertices in each piece of $\mathcal{P}$ and greedily merges pieces until every piece has at least n/k vertices (all from V); let $\mathcal{P}'$ denote the resulting partition of V. Clearly there are at most k pieces in $\mathcal{P}'$, and each has size at most $3^{2/\varepsilon} \cdot \frac{n}{k}$. Thus $\mathcal{P}'$ is a $3^{2/\varepsilon}$-balanced k–partition.

We now upper bound the total cost of all edges removed by the algorithm (over all instances); this also bounds the cost of $\mathcal{P}'$. This is immediate from Claim 5.4 below: since there are t levels and $\mathcal{A}$ achieves an α-approximation to the cost, the total cost is bounded by $\alpha\,t \cdot$ OPT.

Claim 5.4. *For each $i \in \{1, \ldots, t\}$, the sum of optimal values of level i instances is at most* OPT.

Proof. Consider any fixed i, and an instance $\mathcal{I}$ of level i. We will show that the edges of OPT induced on $\mathcal{I}$ form a balanced k_i-partition for $\mathcal{I}$. This suffices to prove the claim since the level i instances partition V (the original vertex-set). Let V' denote the vertices from V in $\mathcal{I}$; note that $|V'| \leq \frac{n_{i-1}}{k_{i-1}} \cdot k_{i-1}^{1-\varepsilon}$. Consider the partition $\mathcal{Q}$ of V' induced by OPT; note that each piece in $\mathcal{Q}$ has size at most n/k. Greedily merge pieces in $\mathcal{Q}$ as long as the size of each piece is at most n/k, to obtain partition $\mathcal{Q}'$; so the number of pieces is:

$$|\mathcal{Q}'| \leq 1 + 2|V'| \cdot \frac{k}{n} \leq 1 + 2\frac{n_{i-1}}{k_{i-1}} k_{i-1}^{1-\varepsilon} \cdot \frac{k}{n} = 1 + 2\,k_{i-1}^{1-\varepsilon} \leq 3\,k_{i-1}^{1-\varepsilon} \leq k_{i-1}^{1-\varepsilon/2} = k_i.$$

The second-last inequality uses $k_{i-1} \geq 3^{2/\varepsilon}$ which is true by the choice of t. Now we can fill pieces of $\mathcal{Q}'$ with dummy singleton vertices of instance $\mathcal{I}$ to obtain a balanced k_i-partition of $\mathcal{I}$. □

This completes the proof of Lemma 5.3. □

References

[AR98]   Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1), 1998.

[AR06]   K. Andreev and H. Racke. Balanced graph partitioning. *Theor. Comp. Sys.*, 39(6):929–939, 2006.

[ARV08]   S. Arora, S. Rao, and U. V. Vazirani. Geometry, flows, and graph-partitioning algorithms. *Commun. ACM*, 51(10):96–105, 2008.

[Bar96]   Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1996.

[BLNZ11]   N. Bansal, K.-W. Lee, V. Nagarajan, and M. Zafer. Minimum congestion mapping in a cloud. In *PODC*, 2011.

[CKN09]   J. Cheeger, B. Kleiner, and A. Naor. A $(\log n)^{\Omega(1)}$ integrality gap for the sparsest cut SDP. In *FOCS*, pages 555–564. IEEE Computer Society, 2009.

[CKR00]   G. Calinescu, H. J. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *J. Comput. Syst. Sci*, 60(3):564–574, 2000.

[CLLM10]   M. Charikar, T. Leighton, S. Li, and A. Moitra. Vertex sparsifiers and abstract rounding algorithms. In *FOCS*, pages 265–274. IEEE Computer Society, 2010.

[CMM06]   E. Chlamtac, K. Makarychev, and Y. Makarychev. How to play unique games using embeddings. In *FOCS*, pages 687–696, 2006.

[CRB09]   N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba. Virtual network embedding with coordinated node and link mapping. In *INFOCOM*, pages 783–791. IEEE, 2009.

[EGK+10]   M. Englert, A. Gupta, R. Krauthgamer, H. Raecke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. In *APPROX*, 2010.

[ENRS99]   G. Even, J. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. *SIAM J. Comput.*, 28(6):2187–2214, 1999.

[FKN03]   U. Feige, R. Krauthgamer, and K. Nissim. On cutting a few vertices from a graph. *Discrete Appl. Math.*, 127:643–649, May 2003.

[FT03]   J. Fakcharoenphol and K. Talwar. Improved decompositions of graphs with forbidden minors. In *6th International workshop on Approximation algorithms for combinatorial optimization*, pages 871–882, 2003.

[GKL03]   A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, October 2003.

[KKS+04]   D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.

[KNS09]   R. Krauthgamer, J. Naor, and R. Schwartz. Partitioning graphs into balanced components. In *SODA*, pages 942–949, 2009.

[KPR93]   P. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC, pages 682–690, New York, NY, USA, 1993. ACM.

[KR11]   R. Krauthgamer and T. Roughgarden. Metric clustering via consistent labeling. *Theory of Computing*, 7(1):49–74, 2011.

[LLR95]   N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2), 1995.

[LM10]   T. Leighton and A. Moitra. Extensions and limits to vertex sparsification. In *STOC*, pages 47–56, 2010.

[LN06]   J. R. Lee and A. Naor. L_p metrics on the Heisenberg group and the Goemans-Linial conjecture. In *FOCS*, pages 99–108. IEEE Computer Society, 2006.

[LR99]   T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[LS10]   J. R. Lee and A. Sidiropoulos. Genus and the geometry of the cut graph. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 193–201. SIAM, 2010.

[MM10]   K. Makarychev and Y. Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *FOCS*, pages 255–264. IEEE Computer Society, 2010.

[Moi09]   A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *FOCS*, pages 3–12, 2009.

[Räc08]   H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, pages 255–264, 2008.

[Rao99]   S. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *Proceedings of the 15th Annual Symposium on Computational Geometry*, pages 300–306. ACM, 1999.

[RS10]   P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *STOC*, pages 755–764, 2010.

[RST10a] P. Raghavendra, D. Steurer, and P. Tetali. Approximations for the isoperimetric and spectral profile of graphs and related parameters. In *STOC*, pages 631–640, 2010.

[RST10b] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. Manuscript, 2010.

[ST04]   Z. Svitkina and É. Tardos. Min-max multiway cut. In *APPROX-RANDOM*, pages 207–218, 2004.

[YYRC08] M. Yu, Y. Yi, J. Rexford, and M. Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *ACM SIGCOMM CCR*, 38(2):17–29, 2008.

[ZA06]   Y. Zhu and M. H. Ammar. Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM*. IEEE, 2006.

Appendix

A   Integrality Gap for SDP Relaxation of Min-Max-Multiway-Cut

Consider the following semi-definite relaxation for the min-max multiway cut problem:

$$\min \quad \lambda$$

$$s.t. \quad \lambda \geq \sum_{(u,v) \in E} ||y_{u,i} - y_{v,i}||_2^2 \qquad\qquad \forall i = 1, 2, \ldots, k \qquad\qquad (10)$$

$$\sum_{i=1}^{k} ||y_{u,i}||_2^2 = 1 \qquad\qquad \forall u \in V \qquad\qquad (11)$$

$$||y_{t_i,i}||_2^2 = 1 \qquad\qquad \forall i = 1, 2, \ldots, k \qquad\qquad (12)$$

$$y_{u,i} \cdot y_{v,i} = 0 \qquad\qquad \forall u \in V, \forall i \neq j \qquad\qquad (13)$$

$$y_{u,i} \cdot \sum_{j=1}^{k} y_{v,j} = ||y_{u,i}||_2^2 \qquad\qquad \forall u, v \in V, \forall i = 1, 2, \ldots, k \qquad\qquad (14)$$

$$||y_{u,i} - y_{v,j}||_2^2 + ||y_{v,j} - y_{w,r}||_2^2 \geq ||y_{u,i} - y_{w,r}||_2^2 \quad \forall u, v, w \in V, \forall i, j, r = 1, 2, \ldots, k \quad (15)$$

$$y_{u,i} \cdot y_{v,j} \geq 0 \qquad\qquad \forall u, v \in V, \forall i, j = 1, 2, \ldots, k \qquad\qquad (16)$$

$$||y_{u,i}||_2^2 \geq y_{u,i} \cdot y_{v,j} \qquad\qquad \forall u, v \in V, \forall u, v \in V, \forall i, j = 1, 2, \ldots, k \qquad\qquad (17)$$

In the above relaxation, each vertex $u \in V$ is associated with k different vectors: $y_{u,1}, y_{u,2}, \ldots, y_{u,k}$. Each of those corresponds to a different terminal. Notice that the last two constraints of the relaxation are the ℓ_2^2 triangle inequality constraints including the origin (the same constraints used by [CMM06]).

As stated, the integrality gap we consider is the star graph which constrains as single vertex u connected by k edges to the k terminals. The value of any integral solution to this instance is exactly $k - 1$ since the only choice is to which terminal should u be assigned. Any choice made results in a min-max objective value of $k - 1$.

Let us construct the fractional solution to the above relaxation. Fix e to be an arbitrary unit vector, and $x_1, x_2, \ldots, x_k$ be k unit vectors which are all orthogonal to e and the inner product between any two of them is $-1/(k-1)$. Set the following fractional solution:

$$y_{t_i,i} = e \qquad\qquad \forall i = 1, 2, \ldots, k$$

$$y_{t_i,j} = 0 \qquad\qquad \forall i \neq j$$

$$y_{u,i} = \frac{1}{k} e + \frac{\sqrt{k-1}}{k} x_i \qquad\qquad \forall i = 1, 2, \ldots, k$$

First, we show that the above fractional solution is feasible. Constraints (11), (12) and (13) are

28

obviously feasible for all terminals. Vertex u also upholds constraint (11) since:

$$\sum_{i=1}^{k} \left\lVert \frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_i \right\rVert_2^2 = \sum_{i=1}^{k} \left(\frac{1}{k^2} + \frac{k-1}{k^2} \right) = 1.$$

Let us verify that vertex u satisfies also constraint (13):

$$\left(\frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_i \right) \cdot \left(\frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_j \right) = \frac{1}{k^2} + \frac{k-1}{k^2} \cdot \left(-\frac{1}{k-1} \right) = 0.$$

Focus on constraint (14). It is easy to verify that for all terminals, the sum of all k vectors that are associated with the picked vertex is exactly e. Since $\sum_{i=1}^{k} x_i = 0$, the sum of all vectors associated with u is also e. Therefore, all constraints of type (14) are satisfied.

Regarding the last three constraints, which are the ℓ_2^2 triangle inequality constraints including the origin, one can verify the following:

$$\lVert e - 0 \rVert_2^2 = 1$$

$$\left\lVert e - \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_i \right) \right\rVert_2^2 = 1 - \frac{1}{k}$$

$$\left\lVert 0 - \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_i \right) \right\rVert_2^2 = \frac{1}{k}$$

$$\left\lVert \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_i \right) - \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_j \right) \right\rVert_2^2 = \frac{2}{k}$$

All the last three constraints are derived from the above calculations. Hence, we can conclude that the fractional solution defined above is feasible for the semi-definite relaxation.

The value of this solution is:

$$\max_{1 \le i \le k} \left\{ \left(1 - \frac{1}{k} \right) + (k-1) \cdot \frac{1}{k} \right\} = 2\frac{k-1}{k}.$$

This gives an integrality gap of $k/2$.

B   Bad Example: Greedy Algorithm for Min–Max k–Partitioning

We show that the naive greedy algorithm that repeatedly uses Small-Set Expansion to remove a part of size n/k performs very poorly. In this example we even assume that there is an exact algorithm for Small-Set Expansion. The graph is a tree on $n = k^2$ vertices $V := \{v\} \bigcup_{i=1}^{k-1} \{u_{i,0}, \ldots, u_{i,k}\}$, and edges $E = \bigcup_{i=1}^{k-1} \{(u_{i,j}, u_{i,j-1}) : 1 \le j \le k\} \bigcup \{(u_{i,0}, u_{i-1,0}) : 2 \le i \le k-1\} \bigcup \{(v, u_{1,0})\}$.

The simple greedy algorithm will cut out parts having small boundary for $k-1$ iterations, namely $P_i = \{u_{i,1}, \ldots, u_{i,k}\}$ for $i \in [k-1]$; note that $\delta(P_i) = 1$ for all $i \in [k-1]$. However the last part $\{v, u_{1,0}, \ldots, u_{k-1,0}\}$ has cut-value $k-1$; so the resulting objective value is $k-1$.

On the other hand, it can be checked directly that the optimal value is at most *four*: Consider the partition obtained by repeatedly taking the first k consecutive vertices from the ordering $v, u_{1,0}, \ldots, u_{1,k}, u_{2,0}, \ldots, u_{2,k}, \ldots, u_{k-1,0}, \ldots, u_{k-1,k}$.